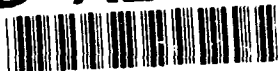
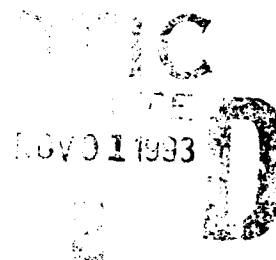


AD-A271 846



②

NAVAL POSTGRADUATE SCHOOL  
Monterey, California



THESIS

SPREADSHEET TECHNIQUES FOR LOGISTICS  
DECISION SUPPORT SYSTEMS

by

Dale Wellington Goeller

June 1993

Principal Advisor

Keebom Kang

Approved for public release; distribution is unlimited.

93-26277



93 10 29 009

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE				
1a Report Security Classification: Unclassified			1b Restrictive Markings	
2a Security Classification Authority			3 Distribution/Availability of Report	
2b Declassification/Downgrading Schedule			Approved for public release; distribution is unlimited	
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)	
6a Name of Performing Organization Naval Postgraduate School		6b Office Symbol (if applicable)	7a Name of Monitoring Organization Naval Postgraduate School	
6c Address (city, state, and ZIP code) Monterey CA 93943-5000			7b Address (city, state, and ZIP code) Monterey CA 93943-5000	
8a Name of Funding/Sponsoring Organization		8b Office Symbol (if applicable)	9 Procurement Instrument Identification Number	
Address (city, state, and ZIP code)			10 Source of Funding Numbers	
			Program Element No	Project No Task No Work Unit Accession No
11 Title (include security classification) SPREADSHEET TECHNIQUES FOR LOGISTICS DECISION SUPPORT SYSTEMS (UNCLASSIFIED)				
12 Personal Author(s) Dale Weinington Goener				
13a Type of Report Master's Thesis		13b Time Covered From To	14 Date of Report (year, month, day) 17 June 1993	15 Page Count 92
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
17 Cosan Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)	
Field	Group	Subgroup	Computer Spreadsheets, Decision Support Systems, Microsoft Excel, Logistics Models	
19 Abstract (continue on reverse if necessary and identify by block number)				
<p>*This thesis considers the use of spreadsheet techniques as a foundation for the development of logistics decision support systems. An inventory model is presented to show the flexibility of spreadsheet techniques and demonstrate the use of various graphical interface techniques. Several distributions of interest in the area of logistics decision making are then modeled, displayed graphically and discussed. Finally a process control model is presented and its implications discussed. Sample spreadsheet coding is provided for all models presented with a primary emphasis on graphic output to aid the logistics manager in decision making</p>				
20 Distribution/Availability of Abstract * unclassified/unlimited — same as report — DTIC users			21 Abstract Security Classification Unclassified	
22a Name of Responsible Individual Keebom Kang		22b Telephone (include Area Code) (408) 626 4035		22c Office Symbol AS/KK

DD FORM 1473.64 MAR

83 APR edition may be used until exhausted

All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

Spreadsheet Techniques for Logistics  
Decision Support Systems

by

Dale W. Goeller  
Lieutenant, Supply Corps, United States Navy  
B.S., University of Missouri at Kansas City, 1979

Submitted in partial fulfillment  
of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL  
June 1993

Author:

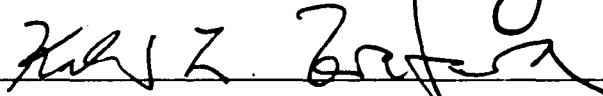


Dale W. Goeller

Approved by:



Keebom Kang, Principal Advisor



K. L. Terasawa, Associate Advisor



David R. Whipple, Chairman  
Department of Administrative Sciences

## ABSTRACT

This thesis considers the use of spreadsheet techniques as a foundation for the development of logistics decision support systems. An inventory model is presented to show the flexibility of spreadsheet techniques and demonstrate the use of various graphical interface techniques. We used several time series models to display the graphical capabilities of spreadsheet programs for decision making. Finally a process control model is presented and its implications discussed. Sample spreadsheet coding is provided for all models presented with a primary emphasis on graphic output to aid the logistics manager in decision making.

Approved For	
CLASS	SECRET
DATE	10/1/81
BY	10/1/81
By	
DATE	
Approved For	
Dist	SECRET
A-1	

SECRETED 5

## TABLE OF CONTENTS

I. INTRODUCTION .....	1
A. BACKGROUND .....	1
B. OBJECTIVE .....	2
C. SCOPE, LIMITATIONS AND ASSUMPTIONS .....	3
D. METHODOLOGY .....	4
E. ORGANIZATION OF THE STUDY .....	5
II. BACKGROUND OF THE PROBLEM .....	6
III. AN INVENTORY MODEL .....	11
IV. SEVERAL TIME SERIES OF INTEREST .....	25
A. INTRODUCTION .....	25
B. AR(1) MODEL .....	26
C. EAR(1) MODEL .....	32
D. M/M/1 QUEUING MODEL .....	36
E. CONCLUSION .....	40

V. PROCESS CONTROL .....	41
A. INTRODUCTION .....	41
B. THE PROBLEM .....	42
C. CONCLUSION .....	47
VI CONCLUDING REMARKS .....	48
APPENDIX A INVENTORY MODEL CODE .....	50
A. MAIN MACRO .....	50
B. INPUT DIALOG BOX .....	64
APPENDIX B AR(1) MODEL .....	66
A. MAIN MACRO .....	66
B. INPUT DIALOG CODE .....	67
C. INPUT WORKSHEET .....	67
D. OUTPUT GRAPHS .....	68
APPENDIX C EAR(1) MODEL .....	71
A. MAIN MACRO .....	71
B. INPUT DIALOG BOX .....	72
C. OUTPUT GRAPHS .....	72

APPENDIX D M/M/1 QUEUING MODEL .....	74
A. MAIN MACRO .....	74
B. WORKSHEET CODE .....	77
APPENDIX E PROCESS CONTROL .....	79
A. MAIN MACRO .....	79
B. WORKSHEET LAYOUT .....	80
LIST OF REFERENCES .....	82
INITIAL DISTRIBUTION LIST .....	83

## LIST OF FIGURES

Figure 1: Input Dialog Box, Inventory Model .....	14
Figure 2: Output Dialog Box, Inventory Model .....	16
Figure 3: Inventory Position, Reorder Point = 18 .....	19
Figure 4: Inventory Position, Reorder Point = 10 .....	20
Figure 5: Net Inventory, Reorder Point = 18 .....	21
Figure 6: Net Inventory, Reorder Point = 10 .....	22
Figure 7: Output Table .....	23
Figure 8: 100 Observations from the AR(1) Model with $\phi=0.9$ .....	27
Figure 9: AR(1) $\phi=0.9$ Batched in size $m=1$ , with 16 batches .....	29
Figure 10: AR(1) $\phi=0.9$ Batched in sizes $m=2$ , $m=32$ , $m=256$ with 16 batches .....	30
Figure 11: 100 Sequential Variables EAR(1) Model with $\phi=0.9$ .....	33
Figure 12: EAR(1) $\phi=0.9$ Batched in size $m=1$ , with 16 batches .....	34
Figure 13: EAR(1) $\phi=0.9$ Batched Values $m=2$ , $m=32$ , $m=256$ .....	35
Figure 14: 100 Observations from the M/M/1 Model .....	37
Figure 15: M/M/1 Batched in size $m=1$ , with 16 batches .....	38
Figure 16: M/M/1 Batched Values $m=4$ , $m=256$ .....	39
Figure 17: Process Control Chart .....	45
Figure 18: Accept / Reject Table .....	45



Figure 19: Result of 1000 Trials .....	46
Figure 20: AR(1) Model, $\phi=0.9$ .....	68
Figure 21: AR(1) Model, $\phi=0.5$ .....	69
Figure 22: AR(1) Model, $\phi=0.0$ .....	69
Figure 23: AR(1) Model, $\phi=-0.5$ .....	70
Figure 24: AR(1) Model, $\phi=-0.9$ .....	70
Figure 25: EAR(1) Model, $\phi=0.9$ .....	72
Figure 26: EAR(1) Model, $\phi=0.5$ .....	73
Figure 27: EAR(1) Model, $\phi=0.0$ .....	73

## **I. INTRODUCTION**

### **A. BACKGROUND**

Spreadsheet programs were probably the most significant applications developed in the early days of the personal computer revolution. Originally designed to imitate and mechanize the accountant's manual spreadsheet calculations; spreadsheets were readily adopted and rapidly gained credibility in the area of Financial Management. By applying the power of computer calculation, in a familiar and time tested format, the financial manager could analyze problems that had previously been prohibitively labor intensive. Eliminating the drudgery of manual calculation allowed the financial manager or accountant to concentrate of the problem at hand and has resulted in a vast array of spreadsheet applications that support the financial manager in his decision making.

The continual development and improvement of spreadsheet packages have further contributed to their importance in financial decision making. Currently available packages offer a broad spectrum of special functions and graphical features that allow increasingly complex analyses to be performed. Three-dimensional spreadsheet packages, just now becoming available on the market, hold even more promise for the future. Realizing the power of

spreadsheet analysis and a certain similarity of issues between finance and logistics, it is logical to consider whether spreadsheets might also be useful in the area of logistics decision making.

The simplicity of modern spreadsheet packages, the availability of both those packages and extremely powerful personal computers, and the increasing sophistication of logistics managers all suggest that an important opportunity is at hand. Current spreadsheet programs contain a variety of specialized functions for the automatic handling of statistical distributions, present value analysis, table analysis, and graphical output. These specialized functions offer the logistician the tools he needs to analyze problems and develop specialized applications to help him make the decisions that he faces every day.

## **B. OBJECTIVE**

Several problems will be modeled using a spreadsheet program and the resulting output will be presented in graphical format as a demonstration of the spreadsheets ability to help the manager visualize the problem and arrive at a decision.

It is recognized that spreadsheet programming may not be the most efficient way to model these individual programs. A specialized program, produced either in-house or purchased from outside vendors, would likely be far superior to a simple spreadsheet program developed by an individual

manager. However the prodigious amount of time required and the high cost associated with professional software development often makes it impossible to produce such packages in a timely manner or for a limited market. The result is that a logistics manager often makes decisions without any form of decision support system, relying instead on intuition, experience and luck.

Spreadsheet programming, however, offers the opportunity for any manager to write programs that can help him make the decisions that confront him on a daily basis. By studying several simple problems, the value of graphical spreadsheet output can be evaluated as a Logistics Decision Support Tool.

### **C. SCOPE, LIMITATIONS AND ASSUMPTIONS**

The primary focus of this thesis is on the potential use of spreadsheet graphical output as a basis for logistics decision support systems. The value of both graphic and tabular output are considered, and various examples are provided to demonstrate the ability of the spreadsheet to help a manager better understand a problem and make better decisions.

Microsoft Excel Version 4.0 was selected for this study because it is one of the newer additions to the list of available spreadsheets on the market and offers a variety of functions, a comprehensive macro language and extensive graphical capabilities. In addition, Excel is reasonably easy to learn and offers a variety of programs to translate its code into other spreadsheet formats.

Coding was not considered particularly important to this thesis and may not reflect optimal or even good coding. The primary intent was to develop code that worked sufficiently well to provide graphical output for consideration. Portions of the code included in the appendices are provided so that others more interested in code generation can have a starting point from which to consider the problems presented. The problems analyzed in the body of this thesis were selected to illustrate the use of spreadsheet graphical output and were not necessarily selected for their individual value in logistics decision making. Problems were primarily drawn from existing literature and are well documented. The data used were generated by spreadsheet models using statistical techniques and may not be applicable to real world scenarios without modification.

#### **D. METHODOLOGY**

This research relied heavily on existing models and theories relating primarily to confidence interval procedures, process control procedures and acceptance sampling procedures. The models developed were adapted from those presented in several papers and implemented as spreadsheet programs.

Though not specifically addressed, the power of current spreadsheet programs is quite apparent. Much of the work on which this paper is based was originally conducted using extensive mainframe computer assets and considerable time in developing suitable output formats. Much of this can now

be replicated, with minimal time and effort, on a personal computer using a relatively simple spreadsheet program. The graphical capabilities of spreadsheet programs allow the decision maker to quickly and easily display the results of his work so that he can better visualize the system he is studying.

#### **E. ORGANIZATION OF THE STUDY**

Chapter II discusses the background of the problem. In Chapter III an inventory simulation model was developed using spreadsheet programming. In Chapter IV several time series models were studied. Chapter V discusses the application of spreadsheets for statistical process control with correlated observations. Chapter VI provides concluding remarks.

## II. BACKGROUND OF THE PROBLEM

Rapid developments in the field of logistics have lead to an increasingly complex environment for the logistics decision maker. Once merely a loose collection of procedures used to support deployed military operations; logistics is now developing into a science with far reaching implications, and applications far beyond the realm of military support. While logisticians were once concerned primarily with moving material they are now faced with deciding what material to move, how to move it, the likelihood it will be required if it is moved and a host of other decisions.

Because of this increase in scope the logistician is now faced with a perplexing array of decisions, many of which are no longer intuitive or experience based. Some form of decision support system must be adopted to help in making timely, efficient and correct decisions. No longer are pencil and paper solutions sufficient to solve the problems presented. The time required for manual calculations could alone render the decision worthless, and the potential for error in such manual calculation could be even worse.

Mainframe computers and calculators offered the first solutions to the problem but neither was well suited to the task. Mainframes were too big, too inflexible, and too complicated for use in other than system wide decision

making. Calculators were portable and easy to use but not sufficiently automated or powerful when dealing with complex logistics problems and concepts that now confront individual managers. Personal computers offered the first realistic hope of providing logisticians with a truly useful and universal platform for the development of decision support systems.

Development of personal computers was insufficient by itself to solve the problem however. The initial personal computers functioned very like the mainframes on which they were modeled. They were slow, not readily available and generally provided cryptic output at best. It was the development of the spreadsheet program that ultimately offered true hope.

Spreadsheets were initially modeled after manual documents used by accountants. Accountants had used the same form for hundreds of years, a ruled sheet of paper with columns and lines that intersected to form little rectangular boxes into which they wrote the information pertaining to the problem. The numbers in the boxes could then be added up, either by column or by row and final answers could be figured out. By automating that simple process on the computer untold hours of calculation and copying were eliminated. Problems that were previously not considered worth the cost of obtaining a solution could now be programmed and easily handled.

Financial managers immediately adapted to the new technology. The format had been designed for them, was familiar and eliminated the drudgery of calculation. Logisticians were then just starting to develop the complicated



ideas to improve the performance of logistics systems. As they worked to change logistics from an art form into a complex science, they rapidly developed a need for automated systems. But the concepts hadn't been developed.

Gradually over the last several years logisticians developed the necessary scientific structure. People were trained in probability, math, calculus, finance and a myriad of other disciplines. The range of problems claimed by logisticians as within their domain increased to the point where complex specialization was essential and some form of computer assistance was required. Unfortunately the rest of the society didn't realize what was going on.

Even the author, with twenty-one years of experience in a variety of military supply and logistics positions, didn't know, until quite recently, that logistics is now concerned with probabilities, failure rates, availabilities, and the many other esoteric ideas logisticians have chosen to adopt. Ordinary people think that logistics is still the art of moving material to support deployed military forces. Consequently no universal logistics software has been forthcoming to solve the problems that confront the logistician. Few people outside the specialty understand that such software is required, after all just moving material doesn't require complex computer techniques.

There have been many programs developed to solve specific logistics problems, but they typically require the user to manipulate the data into a

form acceptable to the particular program and only provide some limited output that may not be of much real use to the decision maker. Custom programs can, and have been, written for various logistics problems but they have traditionally been long laborious projects, written in some highly complex computer language by professional programmers who have little understanding of the purpose of the programs they write. Little has truly been done to provide the logistician with ready access to a broad range of programs essential to his work.

Possibly this problem has finally been solved by the new generation of spreadsheet programs currently available. No longer restricted to one screen of columns that can be acted on by a limited number of functions; the modern spreadsheet provides a broad array of functions, the capability of having multiple interrelated spreadsheets, a simple macro language, and extensive graphics capabilities. The modern spreadsheet may be the software that offers the logistician the power exploit the personal computer just as financial managers have been doing for years.

The logistician now has the tools to develop complex logistics decision support systems on his own. By using simple spreadsheet commands the logistician can now design systems to support him in his decisions as the need for them rises. He can personally take control of the personal computer, without the interference of computer programmers, and develop programs that actually help him make decisions.

Perhaps the most innovative feature of this new generation of spreadsheet programs is the advances that have been made in graphics. Once a tedious process to obtain a barely legible graph; one can now by using a few keystrokes, or a mouse interface, create clear, legible and informative graphic representations with ease. These can be particularly important in analyzing complex logistics relationships, for decision making or for educational purposes. The remainder of this paper will concentrate on the potential of this graphical output as a foundation for Logistics Decision Support Systems.

### III. AN INVENTORY MODEL

The idea of using spreadsheet programming for financial analysis is not new. Procedures have been identified and documented as guides for managers to employ in developing spreadsheet programs to aid them in financial analysis and decision making. In a paper on spreadsheet risk analysis (Seila and Banks, 1990), a detailed example is offered to demonstrate the use of simulation techniques to develop a risk analysis model from an existing spreadsheet through the use of spreadsheet macros.

Simulation is a useful method for studying real world systems that are either too complex or too reliant on random probability for analytical analysis. Simulations, employing models based on the important variables involved, provide considerable insight into the operation of such systems and can serve as a foundation for decision making when more precise methods are not available. While simulation can be performed manually, the complexity of problems that may be studied using manual techniques is extremely limited. Spreadsheet programs and macro techniques offer the manager a powerful tool for analyzing such problems.

Spreadsheet packages offer sophisticated mathematical, statistical and financial functions which the manager can use to develop models of more complex systems. Graphical capabilities can provide important visual aids for

the decision maker as he seeks to identify trends or patterns inherent in such systems. Database functions can provide the manager with ways to catalog and store his results, and macro programming can automate much of the process.

Macros, which are short programs written to accomplish specific tasks, can be used to automate complex procedures and are ideally suited for the execution of the repetitive calculations typical of simulation. Using macro programming a manager can quickly and easily analyze a number of competing scenarios and improve his understanding of a problem. The spreadsheet can thus become a tool to help the manager make better decisions by increasing knowledge of how various systems are likely to perform.

An inventory model was constructed to explore spreadsheet simulation of logistics problems using Microsoft Excel Version 4.0. This problem is typical of those presented in logistics classes and is presented as follows:

(Inventory System with Stockouts and Backorders): A Navy Supply Center (NSC) is planning to install a system to control the inventory of a particular nonrepairable (consumable) item called component XYZ. The time between demands for XYZ is uniformly distributed between 0.2 weeks and 0.8 weeks. The requisition amount of each customer is uniformly distributed between 2 and 6. In the case where customers demand XYZ when it is not in stock (a Stockout), 80 percent must arrange a special order at 100 dollars per order whenever the demand occurs, while the other 20 percent are not urgent, thus will backorder and wait for the next shipment arrival. NSC employs a periodic review-reorder point inventory system where the inventory status is reviewed every four weeks to decide if an order should be placed. NSC's policy is to order up to the stock control level of 72 XYZ's whenever the inventory position, consisting of items in stock plus items on order minus the items on backorder, is found to be less than or equal to the reorder point of 18.

The procurement lead time (the time from placement of an order to its receipt) is constant and requires three weeks. The initial inventory values are 72 units on hand, 0 units on order and 0 units on outstanding backorder. (Modified from Pritsker, 1984, pp.194-195)

Although a somewhat simplified version of reality, this problem is illustrative of the complexity involved in typical logistics problems for which decision support systems are required. There is obviously no single answer that will provide a universal solution. Consequently a suitable decision support system would serve primarily to provide a clear understanding of the processes involved rather than a single answer. Such a system could help a manager make better decisions than might otherwise result.

A simple reading of the problem shows that there are a number of variables, operating together, which affect the inventory at any point in time. By analyzing these variables and the likely relationships between them a model of the system that closely resembles inventory behavior in a real world situation can be developed. A manager can then use a spreadsheet program to code these relationships, automate the calculations involved, and design graphical interfaces to help him understand and make inventory decisions. Code developed for the inventory model can be found in Appendix A.

The primary benefits of such an approach are the virtual elimination of calculation errors, a tremendous reduction in the time required to perform the calculations, and a considerable flexibility in displaying results. Additional benefits include easy modification of the model as new variables are identified,

adaptability of the model to similar type problems, and through the use of macros the ability to provide a user-friendly system for less knowledgeable personnel to use.

One of the features offered in Excel 4.0 is the ability to generate a graphical interface through which users can modify specific program variables and alter program operation without recoding the underlying spreadsheets. This feature, called the dialog box, can be coded into a macro whenever exploration of various options might be of interest to the user. This greatly enhances the users options and allows him to concentrate on the results of the

**Input Sheet**

Item Name:  Run Date:

**Initial Inventory Quantities**

On Hand:  On Order:  On Backorder:

**Requisition Frequency**

Type Distribution:    
Upper Bound:   
Lower Bound:

**Requisition Quantity**

Type Distribution:    
Upper bound:   
Lower Bound:

**Stockout Information**

Type Distribution:    
Upper Bound:   
Lower Bound:   
Cost per Stockout:

**Type of Inventory Replenishment**

System:    
Frequency:   
Q:   
R:

**Procurement Lead Time**

Type Distribution:    
Mean:   
Standard Deviation:

**Time Period to Record Data**

Start Time:  End Time:

Options:

**Figure 1:** Input Dialog Box, Inventory Model

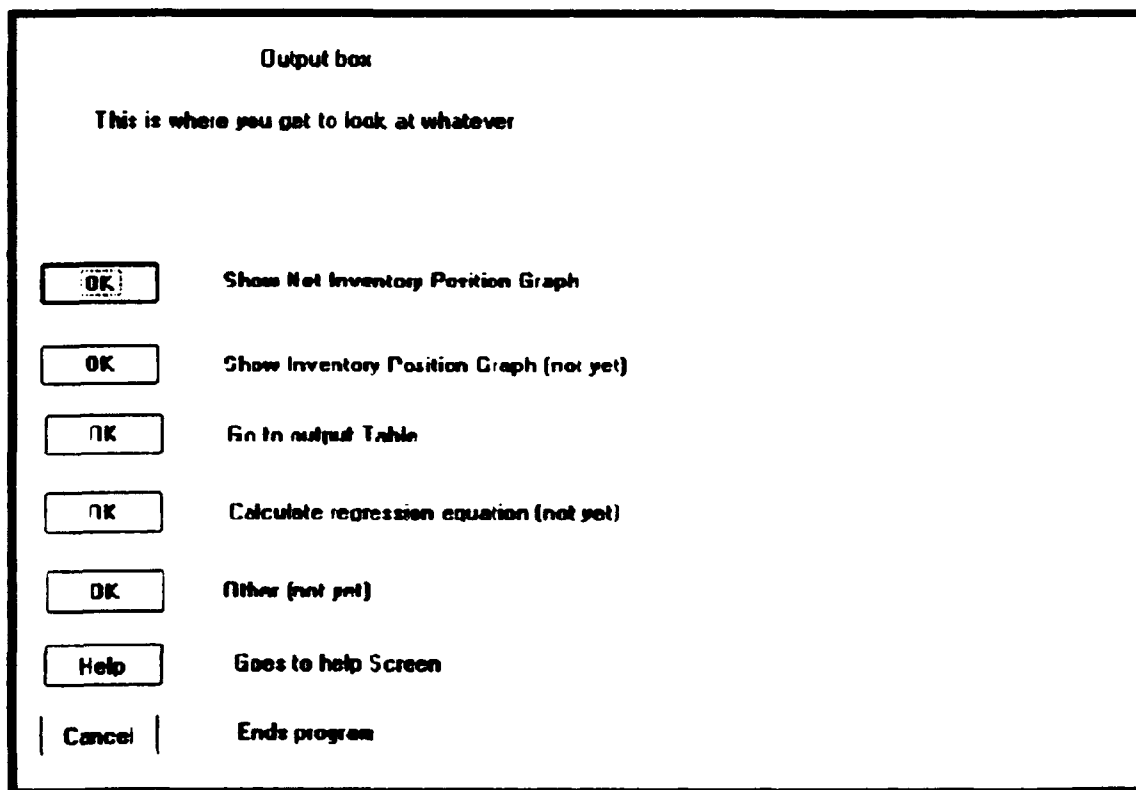
model rather than the code. The dialog box developed for operator input in the inventory model is shown in Figure 1.

As seen in Figure 1, the user has an option to change distributions as well as parameters of the requisition order frequency, the requisition order quantity, stockout information and procurement lead time. The Input Sheet provides two types of inventory system for analysis both the R-System, which bases reorder frequency on time since the last reorder, and a Q-System which bases reordering on inventory level reaching a predetermined level. There is also an option to change the initial inventory values, time period to record and various other values. Other options could be provided, at the discretion of the individual who codes the spreadsheet.

Using a dialog box as an interface for user input allows the spreadsheet programmer an ability to both request information from the user and to verify the format of the users response. The input screen used in the inventory model provides twenty three options which can be varied by a user interested in exploring different scenarios. Various options include modifying the distribution of random variables generated by the simulation, setting initial inventory values, and specifying the type and nature of the inventory replenishment system being considered. There is also a provision for obtaining screens to guide the user in entering specific values and a cancel option to stop further program execution. A similar interface was designed for information output (see Figure 2).



Although not as elaborate as the Input Dialog Box the Output Box provides an equally important interface. The spreadsheet programmer can offer a list of optional outputs which the program can provide. The user can then, by selecting a button on the screen, obtain any of the available outputs desired. Using dialog boxes in this way allows spreadsheet programs to be developed and used by managers with varying degrees of expertise in spreadsheet programming and logistics problem solving.



**Figure 2:** Output Dialog Box, Inventory Model

The coding for each of these dialog boxes is included in Appendix A. While it may appear somewhat complicated it is easily generated using a

separate program provided with Excel 4.0, the Dialog Editor. The Dialog Editor is itself a graphical interface that allows the spreadsheet programmer to design a dialog box on screen, using drag and drop procedures. When the screen is completed it is saved and the coding required is generated by the Dialog Editor. This code can then be called by a macro program when necessary to activate the user interface.

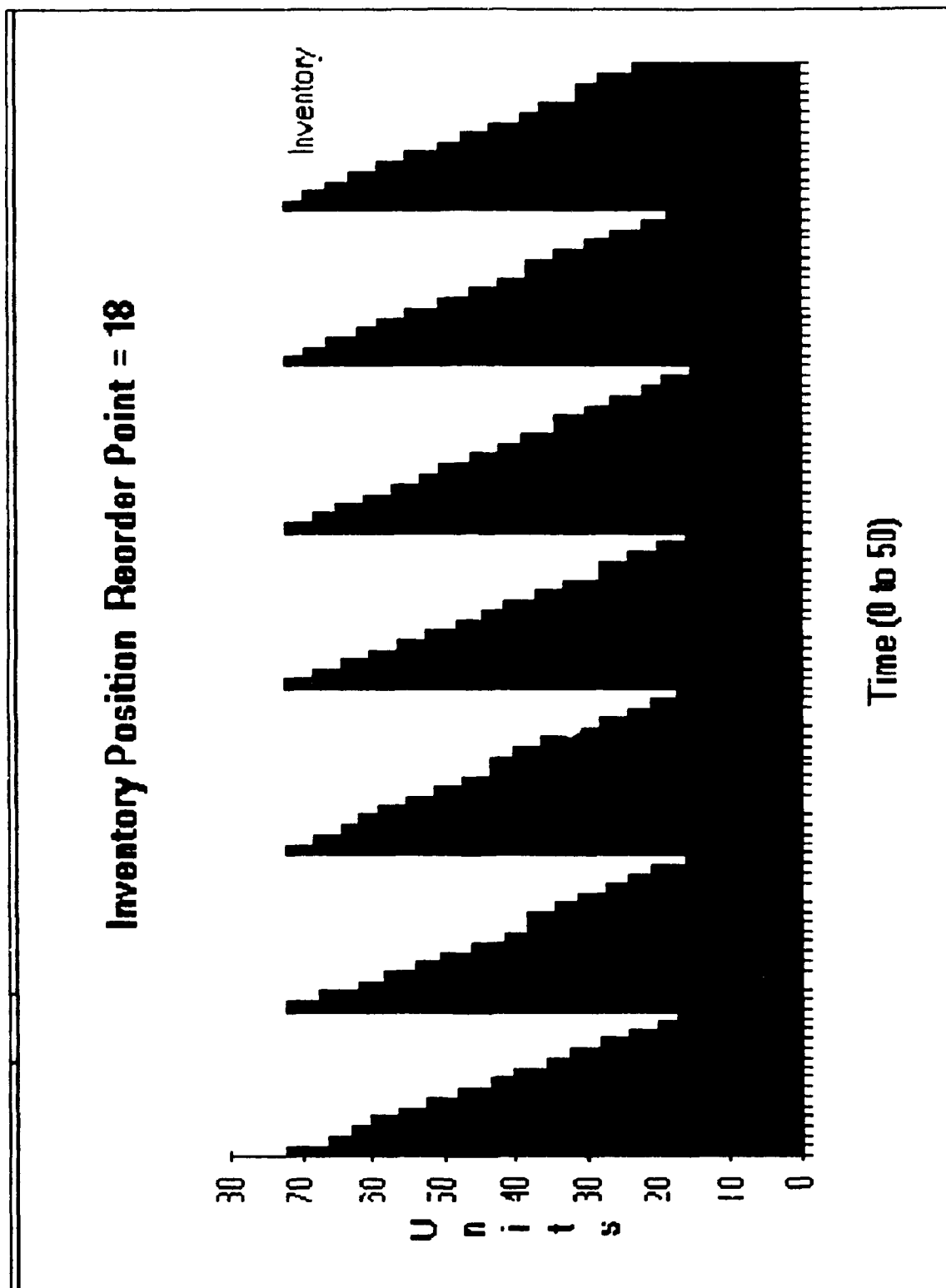
Examples of the graphic output are provided in Figures 3 through 6. These examples show graphs that might be generated by a decision maker who was interested in the effect on inventory levels, backorders and periods of inventory stockout. By generating graphs using two different assumptions concerning the level to which inventory must drop before a reorder is placed the decision maker can compare various scenarios and likely improve his decision making. Figures 3 and 4 graph the Inventory Position (On Hand + On Order - Backorder) and Figures 5 and 6 graph the Net Inventory (On Hand - Backorder) simulated by the model for two different reorder points, of 18 and 10 respectively. All other variables are held constant.

We expect a higher inventory level when the reorder point is 18 than when it is 10, we also expect less stockouts with the higher reorder point. Comparing Figures 5 and 6, we can see that the length of stockouts and the total number of items unavailable when requested is greater with the lower reorder point (Figure 6). Also we can see that the maximum amount of

inventory on hand at the lower reorder point (Figure 6) is typically much lower.

By comparing these 4 graphs a manager can observe a number of important things about how his inventory will react to various reorder points. Together these graphs can help him visualize and understand how the reorder point affects the number, size and frequency of stockouts, the size of orders average inventory and a variety of other details. These graphs don't provide a final solution, but they can be instrumental in providing the manager with the insight he requires to make a decision.

This type of graphical support can provide the decision maker with an idea of what is going on in the system he is simulating. He can see the high and low inventory positions graphically displayed based on the parameters he has specified in his model over a series of trials. Further, he has an option to change the parameters and to run the simulation again to analyze the effects of various changes on the high or low inventory positions he is interested in. This information can help him understand the impact of each variable on his



**Figure 3:** Inventory Position, Reorder Point = 18

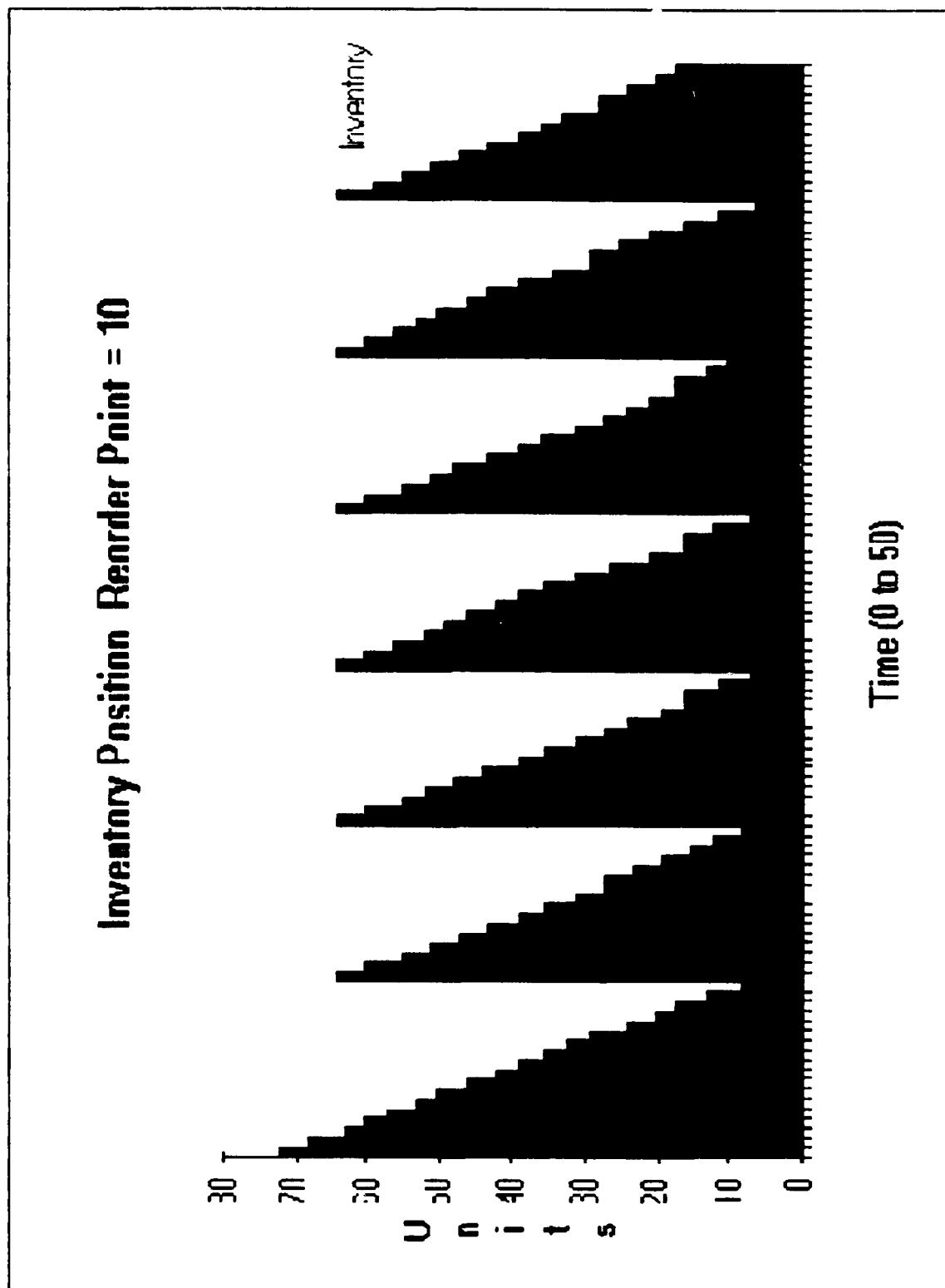
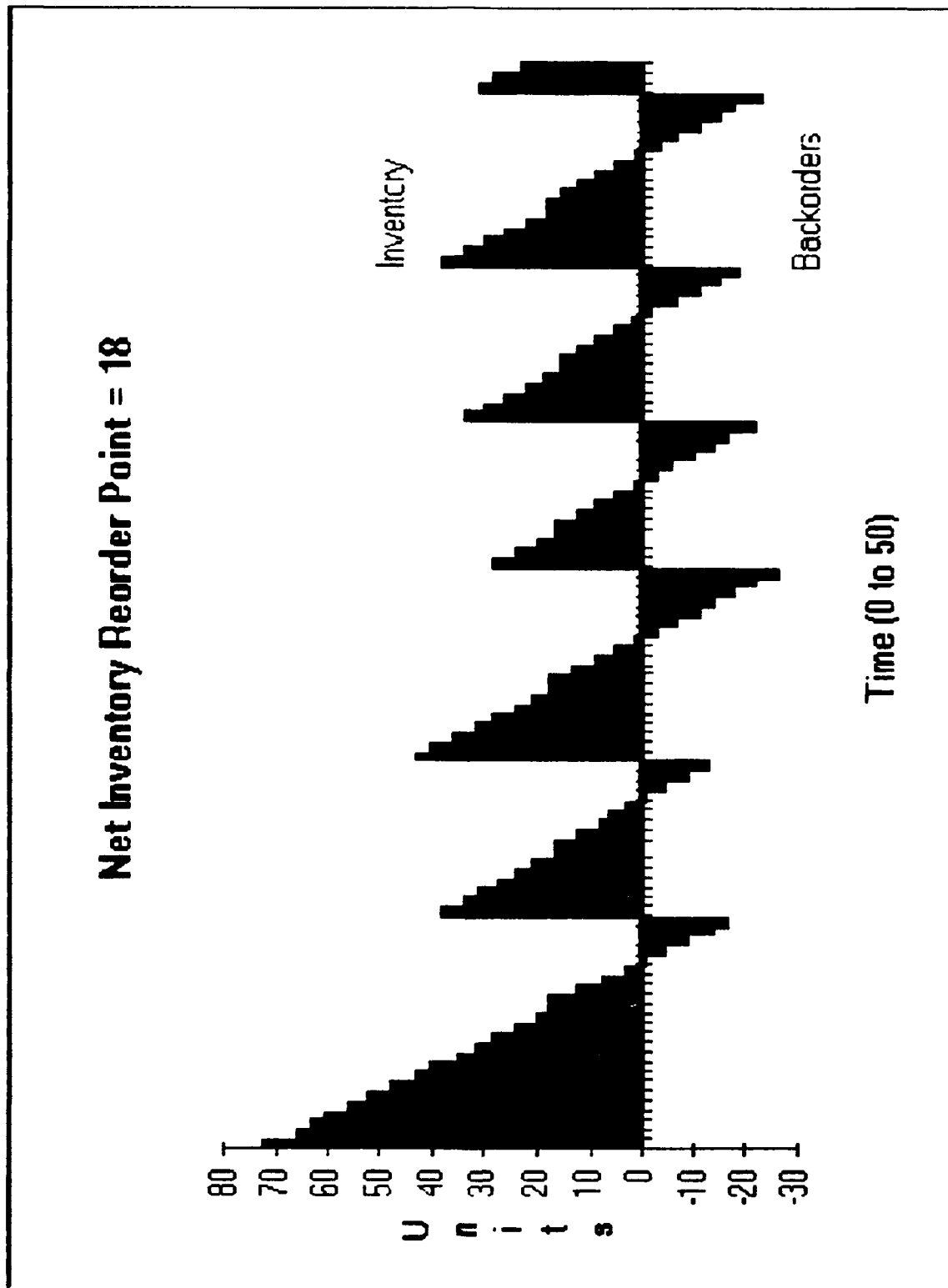
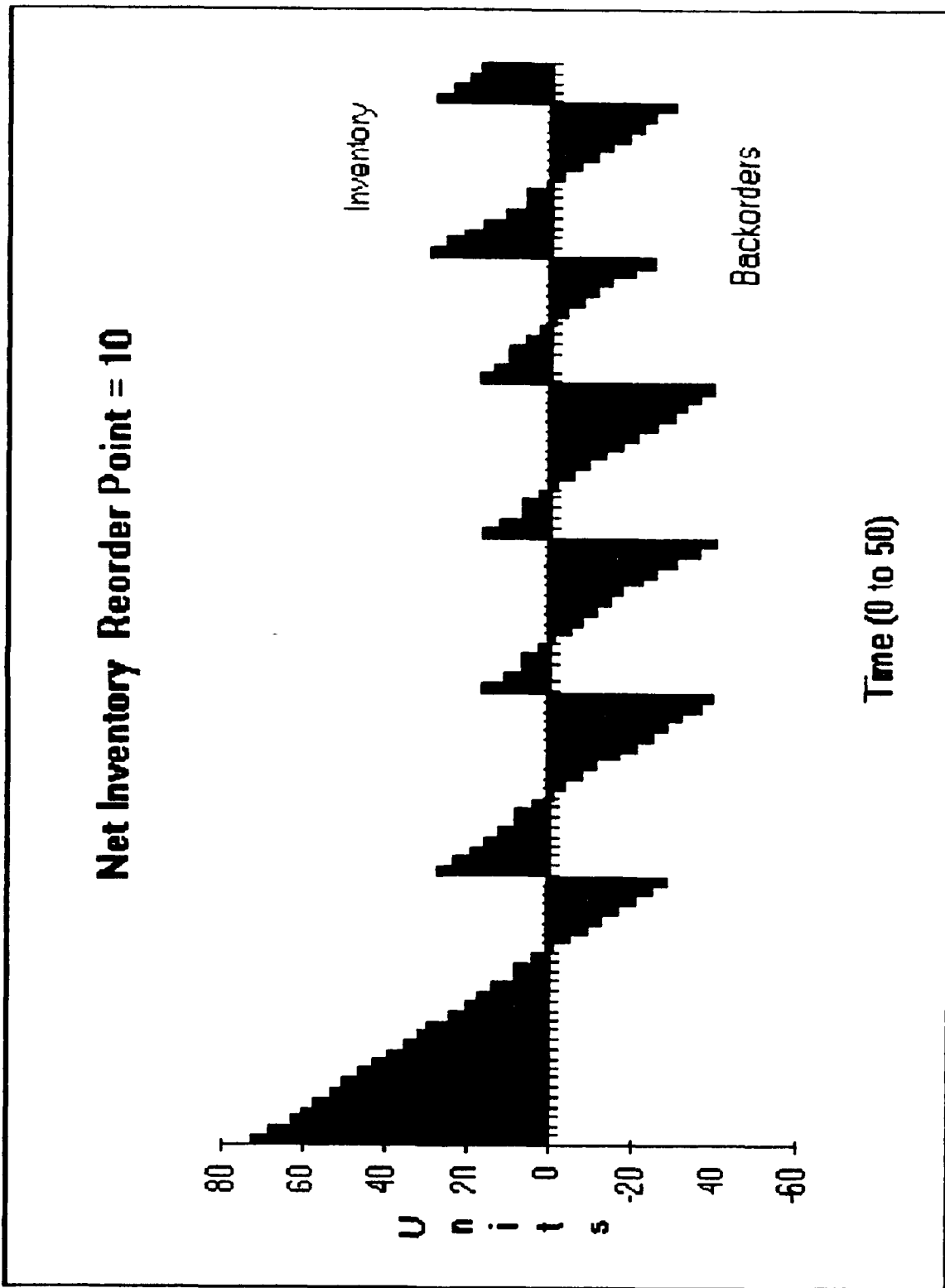


Figure 4: Inventory Position, Reorder Point = 10



**Figure 5:** Net Inventory, Reorder Point = 18



**Figure 6:** Net Inventory, Reorder Point = 10

model, and the real system it portrays. Through such analysis the manager is better able to make decisions related to this type of system.

The decision maker also has ready access a the same data in tabular form as shown in Figure 7.

	Reorder Pt= 18	Reorder Pt=10
Average Customer Requisition Size =	3.8586	3.9697
Number of Reorders =	6	6
Average Requisition Size =	55.5	56.333
Number of Stockouts =	33	71
Average Net Inventory =	13.098	1.75

**Figure 7:** Output Table

By viewing the results in tabular form the manager can consider the specific values involved in greater detail. The simulated average requisition size, number of stockouts, number of reorders, etc. can all be easily determined by reviewing the tabular output. In addition this format output format may be more compatible with the decision making style of some managers. From Figure 7, it is clear that the policy of setting the reorder point to 18, as compared to a reorder point of 10, yields a smaller number of stockouts at the expense of maintaining a higher inventory level.

Spreadsheets offer the ability to manipulate data into forms that are most useful to an individual decision maker. In any case the value of spreadsheet analysis of logistics problems is significant. A complex problem can be simulated or actual data from historical observation can be input and the



simulation portions of the model bypassed. The resulting data can then be analyzed or displayed rapidly by the manager using spreadsheet commands to group, organize or present the data in ways that can provide meaningful insight and help improve decisions .

## IV. SEVERAL TIME SERIES OF INTEREST

### A. INTRODUCTION

We will now consider applying spreadsheet techniques to a less straightforward logistics concept. Statistical analysis often demands the assumption of independent, identically distributed (i.i.d.) random variables. Often this is not the case. Logisticians frequently deal with such variables as units produced on only one machine or component failures occurring in one particular type of aircraft. Are these events independent? In fact these events may be correlated. Either the machine wears at a constant rate or the aircraft characteristics impose a particular stress on a given part. Any of a myriad of details can influence the random nature of the observed part. If the systematic error involved is significant the rule of independence is violated and the system under consideration may not be amenable to a simple statistical analysis.

This problem has been frequently discussed in the literature, but except in an academic environment such readings are seldom high on a practitioners' reading list. It is much simpler to assume that all variables are both independent and identically distributed and more restrictive cases are seldom considered at the practical level. However, by applying spreadsheet technology to this notion it is easy to see problems associated with this simplification.

Using ideas developed in Kang and Schmeiser (1990), a graphical analysis of several correlated processes will be considered.

We will use three different time series models to understand the difficulties in analyzing correlated data: AR(1) model, EAR(1) model and M/M/1 queuing model. By using simple spreadsheet techniques each of these situations can be easily modeled and the results graphically displayed, results which will immediately convince even the most skeptical observer of the potentially undesirable results obtained if an erroneous assumption of independence is employed.

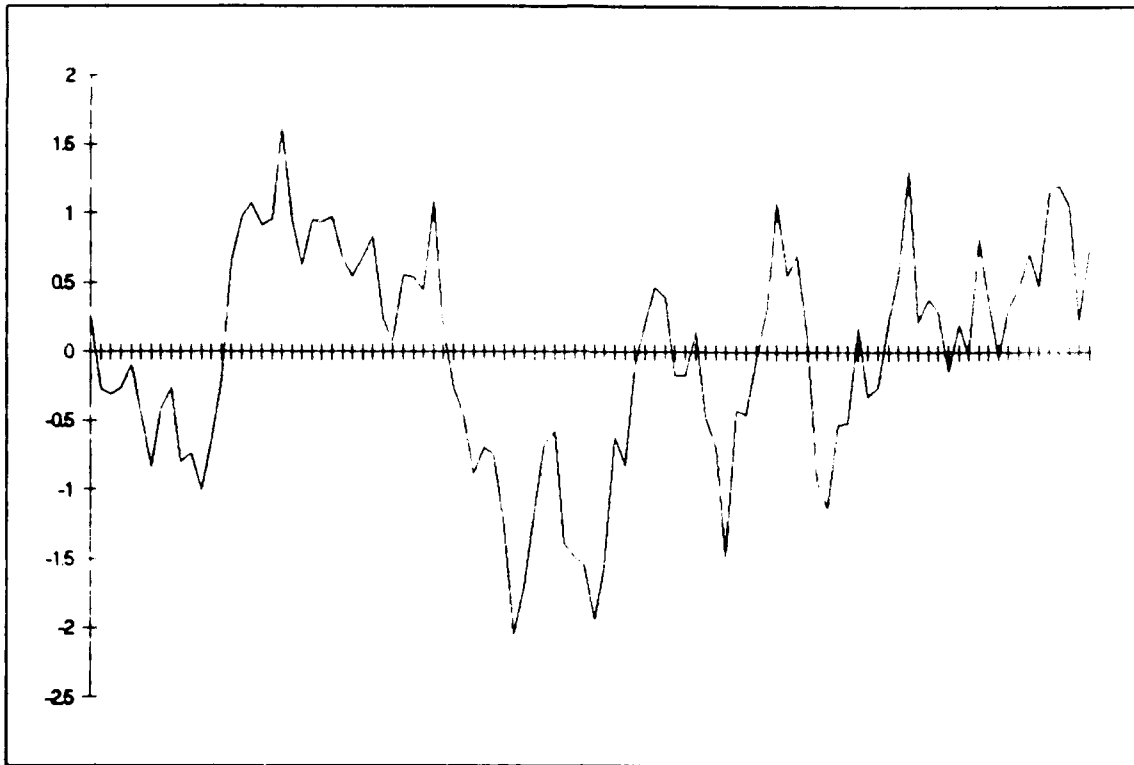
## B. AR(1) MODEL

The AR(1) process,  $X_t$ , can be defined by the equation:

$$X_t = \mu + \phi (X_{t-1} - \mu) + \epsilon_t \quad (1)$$

where  $\epsilon_t \sim N(0, 1 - \phi^2)$ . Without loss of generality, we set  $\mu = 0$ . Note that AR(1) process with  $\phi = 0$  becomes normal i.i.d. process. As  $\phi$  approaches to 1 from 0 higher correlation will occur. The steady state AR(1) process can be generated by setting  $X_1 \sim N(0, 1)$  and recursively generating  $X_2, X_3, \dots$

Using a spreadsheet, graphic representations of such a system can be generated to provide a visual presentation of the typical distribution of such a process. Figure 8 represents a series of 100 sequential values generated using Equation (1), with  $\phi=0.9$ .



**Figure 8:** 100 Observations from the AR(1) Model with  $\phi=0.9$

A  $100(1-\phi)\%$  Confidence Interval can be constructed, using the equation:

$$\bar{X} \pm t_{1-\alpha/2, n-1} \left( \frac{S}{\sqrt{n}} \right) \quad (2)$$

where  $\bar{X}$  is the sample mean, defined as  $\bar{X} = \sum_{i=1}^n X_i / n$  ; and S is the

sample standard deviation, defined as  $S^2 = \sum_{i=1}^n (X_i - \bar{X})^2 / (n-1)$  ;  $t_{1-\alpha/2, n-1}$  is the

$(1-\alpha/2)$ th quantile of the Student's-t distribution with  $n-1$  degrees of freedom;

and  $n$  is the number of observations in the sample. The quantity  $S/\sqrt{n}$  is referred to as the *standard error*.

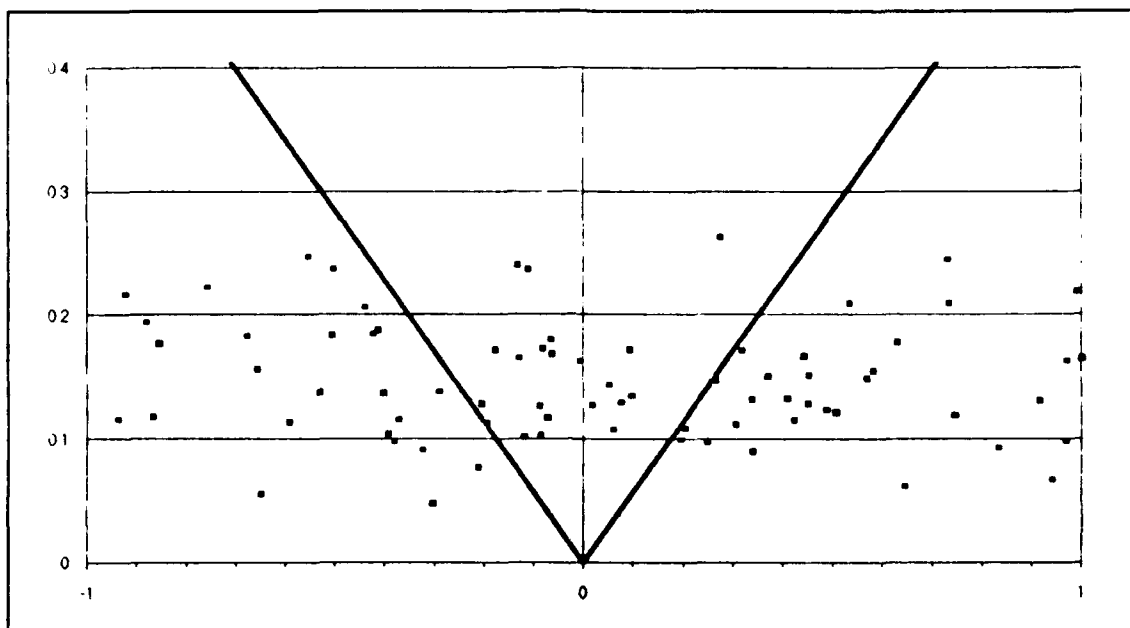
To further explore the properties of an AR(1) model a number of points are generated and graphed following a procedure described in Kang and Schmeiser (1990). A spreadsheet program was created to generate 4,096 sequential values. These values are then grouped into successively larger batches with the resulting mean value of each batch being considered as representative of that batch. The first 16 mean values, for each batch size, are then used to represent 16 samples of that given batch size drawn successively from an ongoing process.

This procedure is replicated for 100 times. The confidence interval can be calculated using Equation (2), and the coverage for a given nominal coverage  $1-\alpha$  (we use  $\alpha=0.1$  as an example) is estimated by the proportion of the points that lie above the lines (Kang and Schmeiser, 1990),

$$y = \frac{|\bar{X} - \mu|}{t_{1-\alpha/2, n-1}} \quad (3)$$

where  $\mu=0$  in this case. In Figure 9, 100 replications of 16 batches each of which has a batch size 1 (i.e.  $m=1$ ) were simulated for the AR(1) process with  $\phi=0.9$ . The coverage is estimated by the proportion of the points that fall above the V-shape lines calculated from Equation (3) that is shown in the figure. The

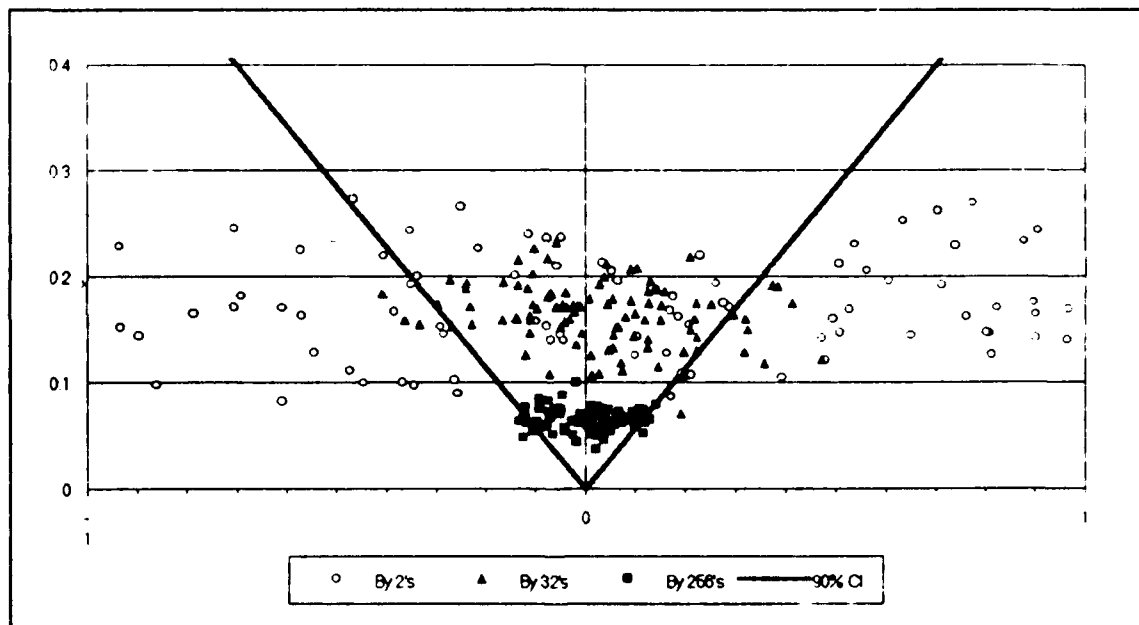
graph shows that the values are widely dispersed and that only 21% of the confidence intervals actually cover the true mean  $\mu$ , while the nominal coverage is 90%. As mentioned before, this poor coverage is due to the correlation between observations.



**Figure 9:** AR(1)  $\phi=0.9$  Batched in size  $m=1$ , with 16 batches

When the process contains positive correlation as can be observed in this case of AR(1) model with  $\phi = 0.9$ ,  $S^2$ , defined in Equation (2), is not an unbiased estimator of  $\sigma^2$ , the process variance. In fact,  $E(S^2) < \sigma^2$ . This means that the confidence interval generated from Equation (2) is narrower than it should be; thus the coverage of the confidence interval is much lower than the nominal value. One way to overcome the effects of correlated data is to batch the observations.

The next graph demonstrates the effect of batching the observations in successively larger groups for the case of  $\phi=0.9$ . Individual observations are first grouped into batches of  $m$  units each. Figure 10 displays batch sizes where  $m=2, 32, 256$  observations, the mean value of each of these groups is then used to represent that group. 100 replications of sixteen such observations are made and the resulting means and standard errors are plotted for each value of  $m$ .



**Figure 10:** AR(1)  $\phi=0.9$  Batched in sizes  $m=2, m=32, m=256$  with 16 batches

The graph presented in Figure 10 visually provides the manager with insight into the effects of batching correlated data. The value of the mean is plotted on the x-axis against the value of the resultant standard deviation on the y-axis. One can see that the data when grouped by 2's is still widely dispersed and coverage is poor. As the batch size increases an obvious pattern

starts to become clear, coverage improves as the values cluster within the confidence interval and the standard error values decrease as the sample size increases. Eventually we deserve the pattern that we expect from i.i.d. normal process.

The decision maker can use such graphical presentations to help him identify the trends involved and use them to improve his decision making. By batching the individual data points the correlation of the data diminishes and is reflected in the clustering of the data for batches of increasing size.

Graphs for four additional cases using the AR(1) model can be found in Appendix B. The graphs present the effects on correlation for various values of  $\phi$ . The cases include  $\phi=0.9$ ,  $\phi=0.5$ ,  $\phi=0.0$ ,  $\phi=-0.5$ , and  $\phi=-0.9$ . The graphic presentation in each case is similar to that presented above. By analyzing and comparing the individual graphs a manager can begin to understand the significance of the correlation in his data.

The cases where  $\phi=-0.5$  and  $\phi=-0.9$  show the effect of negative correlation of data. While the values generated in these cases, even for small batch sizes, tend to provide higher than nominal coverage, the standard error is somewhat smaller than for the positively correlated cases, where  $\phi$  is greater than 0.

The AR(1) model with  $\phi=0.5$  shows improved coverage at all batch sizes when compared with the  $\phi=0.9$  example shown above. This is because  $\phi$  is smaller and consequently each successive data point is less correlated to its predecessor than in the 0.9 case. The improvements in coverage afforded by



batching the data continue to apply and the graphs clearly demonstrate that point.

The AR(1) model with  $\phi=0.0$  is, in fact the Normal (0,1) case. By reducing the correlation value to zero there is no correlation with the preceding values and the graphs depict the relationship that would be expected under the assumptions of the classical model; the coverages are approximately the same as the nominal value (90% in this case), and the standard error decreases as  $m$  (batch size) gets larger, at a rate of  $1/\sqrt{m}$ .

The spreadsheet designed to provide the graphical output in this section can be found in Appendix B. Macro coding is used to replicate the calculations of 4,096 individual variables 100 times in the generation of each graph presented. Each run consists of generating the individual values, grouping those into sequential samples of various sizes, and plotting the mean and standard deviation of sixteen like samples on a graph. Spreadsheet programming has thus condensed millions of calculations into simple visual presentations that can have a major impact on the decisions confronting a manager dealing with such correlated data.

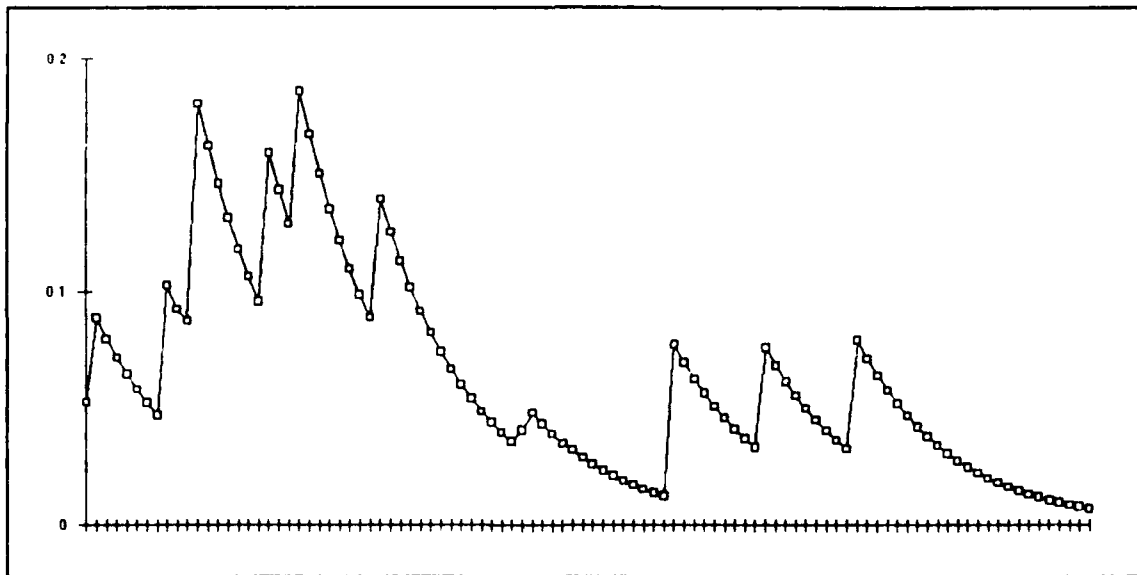
### C. EAR(1) MODEL

The EAR(1) process is defined as:

$$\begin{aligned} X_t &= \phi X_{t-1} && \text{with probability: } \phi \\ &= \phi X_{t-1} + \epsilon_t && \text{with probability: } 1-\phi \end{aligned} \quad (4)$$

where  $\epsilon_t \sim \text{i.i.d. Exp}(1)$ .

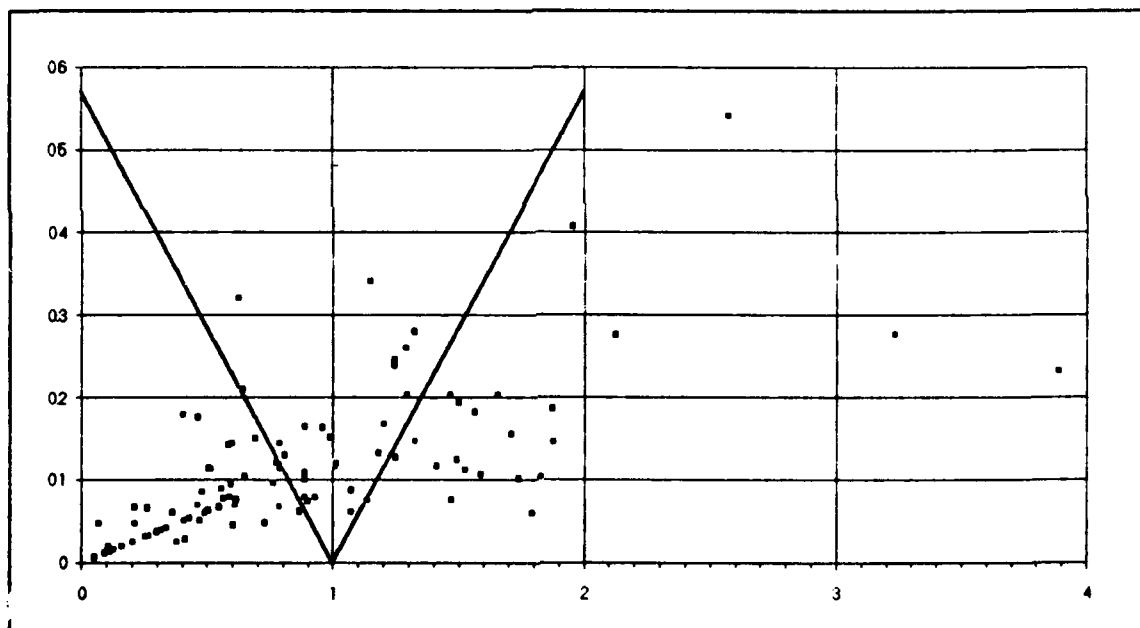
The EAR(1) model has the same autocorrelation structure as in the AR(1) model, except non-normal (exponential) error terms are added. It is worse than the AR(1) model in terms of coverage for a given value of  $\phi$ . This is due to asymmetric error terms. The resultant variables show a steady pattern



**Figure 11:** 100 Sequential Variables EAR(1) Model with  $\phi=0.9$

interrupted periodically by sudden and dramatic jumps. This type of pattern is representative of situations such as machinery repair in which the operation of a machine steadily declines until maintenance or calibration is performed and the pattern of wear or maladjustment starts over again. Figure 11 provides an example of this type of pattern based on 100 sequential observations using  $\phi=0.9$ .

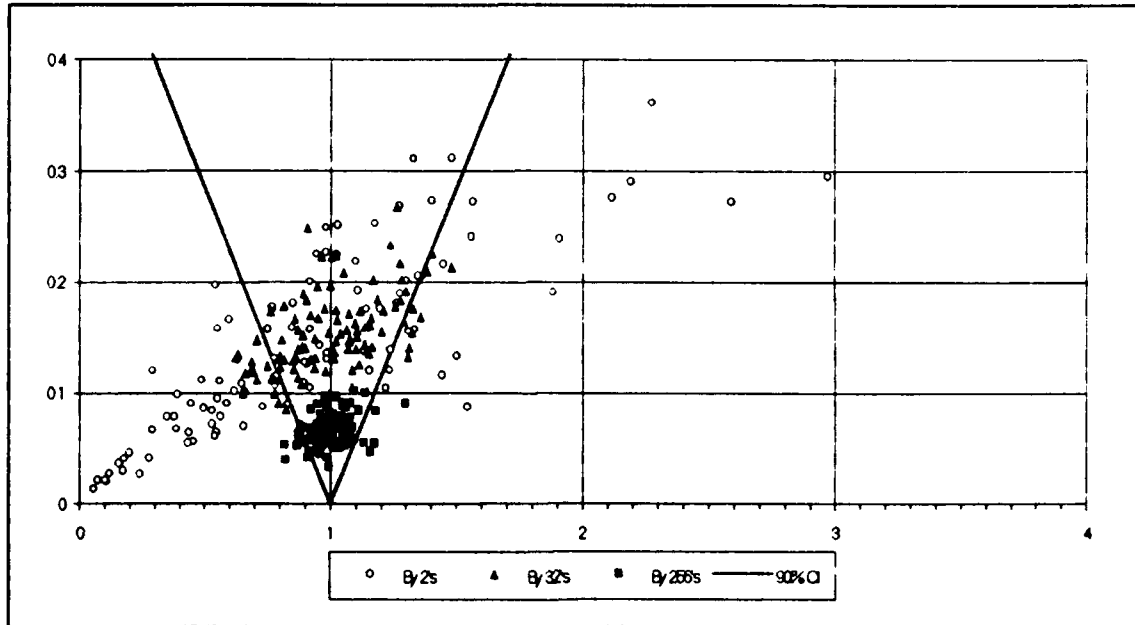
To further explore the properties of an EAR(1) model a number of points are again generated and graphed as in the AR(1) model above. Figure 12 shows 100 points, each representing the sample mean and sample standard deviation of a series of 16 sequential values individually generated from Equation (4). This graph shows that the values are widely dispersed and highly asymmetric. Again the coverage of the true mean ( $\mu=1$  in this case) is much smaller than the nominal value 0.90. In fact only 25% of the confidence intervals actually cover the true mean.



**Figure 12:** EAR(1)  $\phi=0.9$  Batched in size  $m=1$ , with 16 batches

Batching can help overcoming both the effects of data correlation and non-normality inherent in the EAR(1) model. The next graph demonstrates the effect of batching the values in successively larger groups for the case of  $\phi=0.9$ . Individual observations are first grouped into 16 batches of size 2, 32, and 256

the mean value of each of these groups is then used to represent that group. 100 replications of sixteen such observations are made and the resulting means and standard errors are plotted in Figure 13.



**Figure 13:** EAR(1)  $\phi=0.9$  Batched Values  $m=2$ ,  $m=32$ ,  $m=256$

Again the decision maker can easily see from the graphical presentation that as the batch size,  $m$ , is increased the resulting values tend to cluster more tightly and the coverage improves. The individual points plotted display greater symmetry and the standard errors are progressively reduced with each increase in batch size, leading eventually to the acceptability of using classic confidence interval procedures to describe the behavior of some of the larger size batches.

The spreadsheet programs and additional graphs for the EAR(1) model can be found in Appendix C. The spreadsheet programming for the EAR(1)

model is virtually identical to that in the AR(1) model. According to Sargent, Kang and Goldsman (1992), the EAR(1) model requires larger batch size to satisfy the assumptions required for classical confidence interval procedures, than the AR(1) model due to its non-normal error terms. The additional graphs provided consider the cases where  $\phi=0.9$ ,  $\phi=0.5$  and  $\phi=0.0$ . EAR(1) model with  $\phi=0.0$  is the i.i.d. Exp(1) case.

#### D. M/M/1 QUEUING MODEL

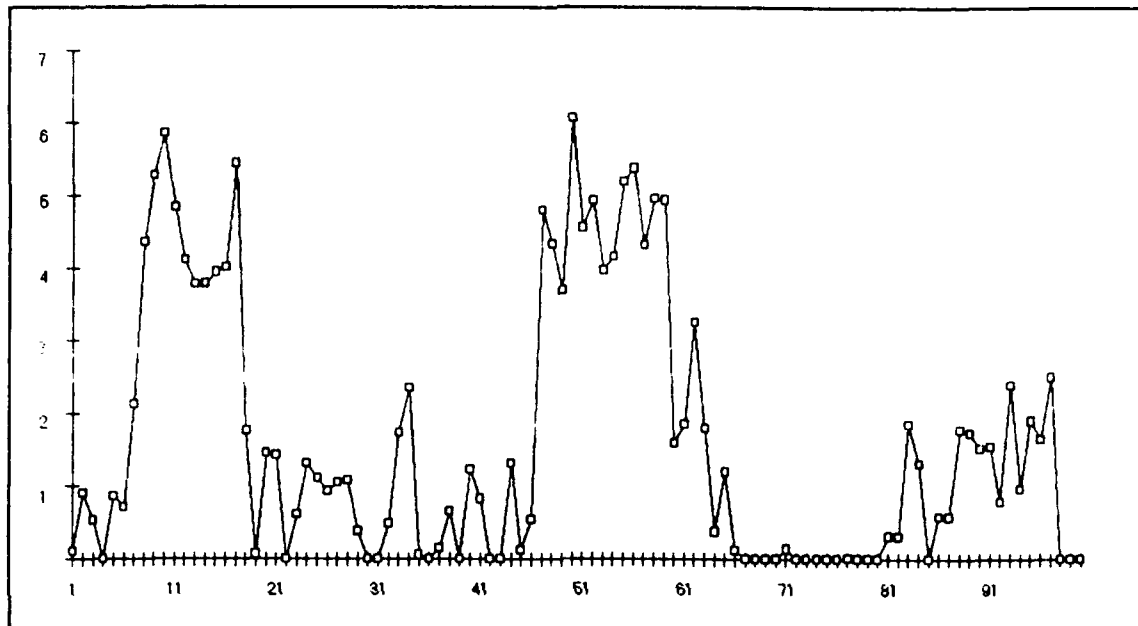
The M/M/1 queuing model represents a single channel waiting line model with Poisson arrivals and exponential service times. The model is designed to study the waiting time for successive customers in the queue. The following equations (from Anderson, Sweeney, Williams, 1991) specify this relationship:

$$W_q = \frac{L_q}{\lambda} \quad (4)$$

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} \quad (5)$$

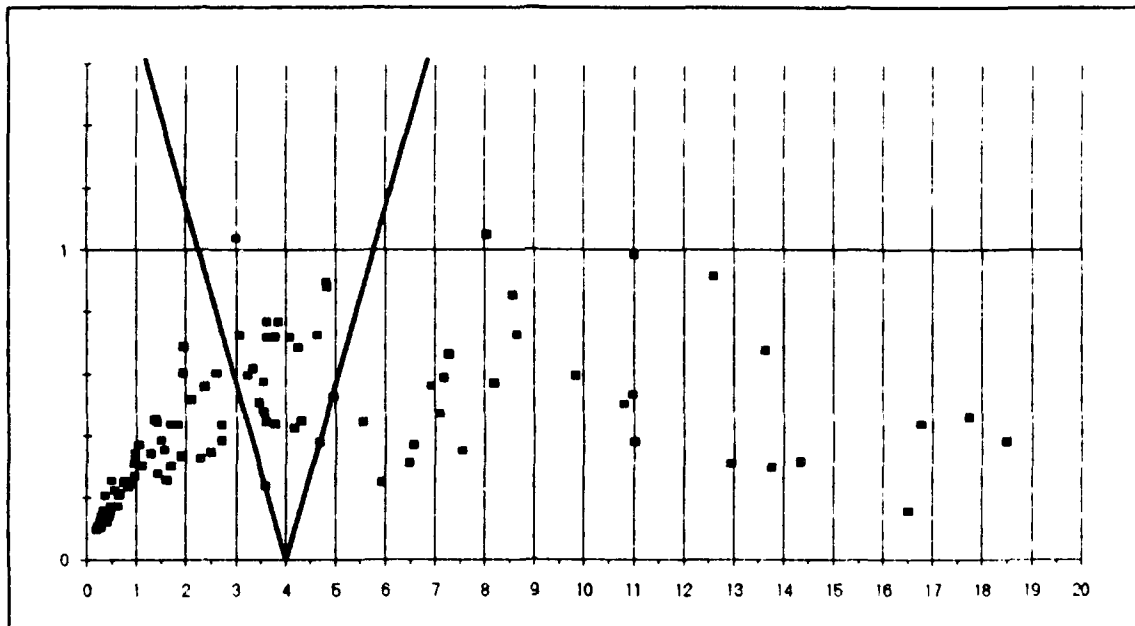
We used the values,  $\lambda=0.8$ ,  $\mu=1.0$  for this case. In a system such as this the values generated for  $W_q$  are obviously dependent on values immediately preceding. Balking is not considered and the model confines itself to only those customers that actually join the queue. A graphic depiction of the situation is presented in Figure 14, where the horizontal axis represents each individual

customer that sequentially enters the waiting line and the vertical axis shows the customer waiting time.



**Figure 14:** 100 Observations from the M/M/1 Model

As shown in Figure 14 the variables generated in an M/M/1 process appears to be very dependant on their predecessors. In fact such a system is considered to be both highly positively correlated and non-normal. Again we plot 100 points, each representing the sample mean and sample standard error of a series of 16 sequential values individually generated from Equations (4) and (5).

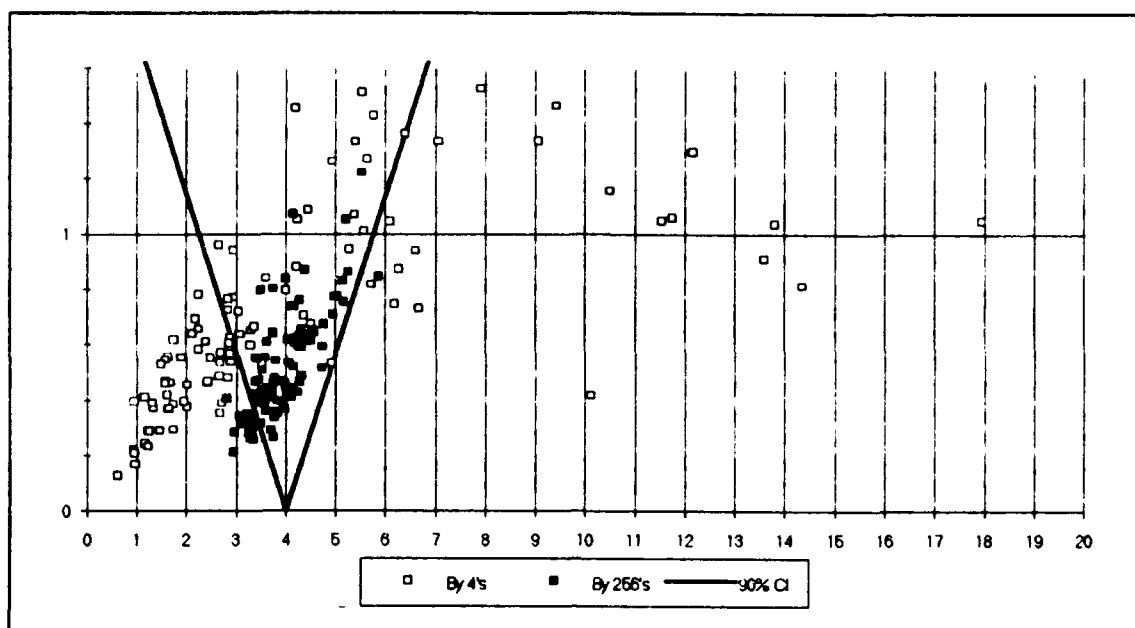


**Figure 15:** M/M/1 Batched in size  $m=1$ , with 16 batches

Figure 15 shows the values to be widely dispersed, and asymmetric. The coverage is poor, only 22 confidence intervals generated from the 100 replications actually cover the true mean ( $W_q=4$  in this case). Again this poor coverage is the result of correlation, the waiting time of any individual customer is dependant on both the number of customers in line ahead of him and the length of time required to service each of those customers.

Once again batching can help overcoming both the effects of data correlation and non-normality, even when the effects are quite large as in the M/M/1 model. The next graph demonstrates the effect of batching the values in successively larger groups. Individual observations are this time grouped into only two batches of 4 and 256 observations, the mean value of each of these groups is then used to represent that group. Sixteen such observations

are made and the resulting mean and standard deviation are plotted in Figure 16.



**Figure 16:** M/M/1 Batched Values  $m=4$ ,  $m=256$

Again the graphic presentation makes it easy to observe the improvements in coverages due to increasing batch size. The individual points plotted display greater symmetry in the case of batch size 256, compared to a case of batch size 4, and the standard errors are considerably reduced.

The spreadsheet programs used in the M/M/1 model can be found in Appendix D. Coding for the M/M/1 model is somewhat different from the two preceding examples and execution is somewhat slower. The same spreadsheet and graphical output formats have been maintained, and thus the resulting output is easily compared with that generated for the AR(1) and EAR(1) cases.



## **E. CONCLUSION**

The three models discussed in this chapter are representatives of a number of problems that the logistics decision maker may face. Often these problems are treated as if the variables involved were independent. While this undoubtedly the case for some problems, it is not always true.

Unrecognized correlation of data can seriously affect the validity of a managers decisions. Such correlation can not be eliminated, because it is a reflection of the natural processes and inherent characteristics of systems. The problem facing decision makers is how to identify and deal with such correlation.

Classical confidence interval procedures are valid only if independent, identically distributed normal assumptions are satisfied. This may not always be the case. Various logistics problems such as turn around time for equipment repair, warehousing and transportation problems all may be highly correlated.

Spreadsheet analysis and graphical presentation can help the decision maker visualize such systems and recognize the consequences involved. While the processes discussed in this thesis are all based on simulated data spreadsheet programming is not limited to such simulations. Any historical data or physical sampling procedure can provide the raw data for spreadsheet analysis. When this data is available it can provide important insight into the decision makers real world problems.

## V. PROCESS CONTROL

### A. INTRODUCTION

Process control is another area in which graphical spreadsheet analysis shows considerable promise. Items successively produced in an ongoing production process are not likely to be totally independent. As discussed in Chapter IV, they will exhibit both random and systematic variability dependant on the characteristics of the production system involved.

This systematic variation can become quite problematic in the area of acceptance sampling. Sampling plans have been calculated, published and standardized. Unfortunately, these standard plans generally assume independent, identically distributed observations in their construction. Application of such plans when the processes involved are not in fact independent can lead to serious consequences. In particular it increases the risk of making a wrong decision based an ill suited sampling plan.

Type I error is considered to be the rejection of a good lot during the sampling process and Type II error is the acceptance of a bad lot. Type I error is considered to be *Producer's Risk* as the producer bears the cost of choosing to reject a good lot. Type II error is considered to be the *Consumers Risk* because it is the chance that a bad lot will not be detected by the sampling

plan employed. Attempts to balance acceptable risk levels of each type further compound the problem.

In some situations high producer's risk is acceptable or even desirable because of serious repercussions that could occur from passing even 1 bad lot. In cases such as weapons manufacture or the production of nuclear power components the danger of a component not working properly is so significant that the consumer's risk must be reduced till it is virtually non-existent. This is a costly proposition for the producer, but it is generally considered more desirable than the alternative.

The opposite situation is equally likely to exist however. For cheap, disposable, harmless items the consumer is often willing to put up with more risk at his level. If a fifty cent ink pen doesn't work properly he is willing to buy another that may or may not work any better, rather than buy a two hundred dollar Mount Blanc model that he is sure will work as perfectly.

All this consumerism and logic is however wasted if the acceptance sampling plan in use does not properly account for the process it is intended to monitor.

## **B. THE PROBLEM**

In an ongoing production process it is often very likely that there is a correlation between successive items produced. Standard acceptance-sampling plans are usually not designed to deal with such correlation and consequently

can result in costly and time consuming errors. Computing the probabilities required to design sampling plans for general correlated processes is often complex, and sometimes intractable. In his paper, Nelson (1990) proposed a methodology for estimating single-sampling attribute plans for any production process that can be simulated.

In a typical single sampling plan of  $(n,c)$ ,  $n$  samples are chosen for inspection, and  $c < n$  is the acceptance number such that the lot is declared acceptable if no more than  $c$  defective items are discovered in the sample. The values of  $n$  and  $c$  are chosen to provide a prespecified *producer's risk* and *consumer's risk*. In this chapter, we develop a spreadsheet program to understand the pitfalls of using standard sampling plans when the process in fact is correlated.

First we start with i.i.d. measurement (i.e.  $\phi=0.0$ ) for which standard acceptance sampling plans are appropriate. We use process  $X$  as the in-control-process that has the normal distribution with a mean of 10 and the variance 1. The process  $X'$  is considered an out-of-control process that has the same variance as  $X$ , but a process mean that has shifted to 11.294.

For such processes a defective item is defined as an item from the process that is beyond the tolerance limits. In this example we chose the upper tolerance limit  $\tau_U=12.5758$  and the lower tolerance limit  $\tau_L=7.4242$  as shown in Table 1. These numbers are chosen so that the producer's risk is 0.01 and the consumer's risk is 0.10: i.e.

$$Pr (\tau_L < X < \tau_U) = 0.99 \quad \text{and}$$

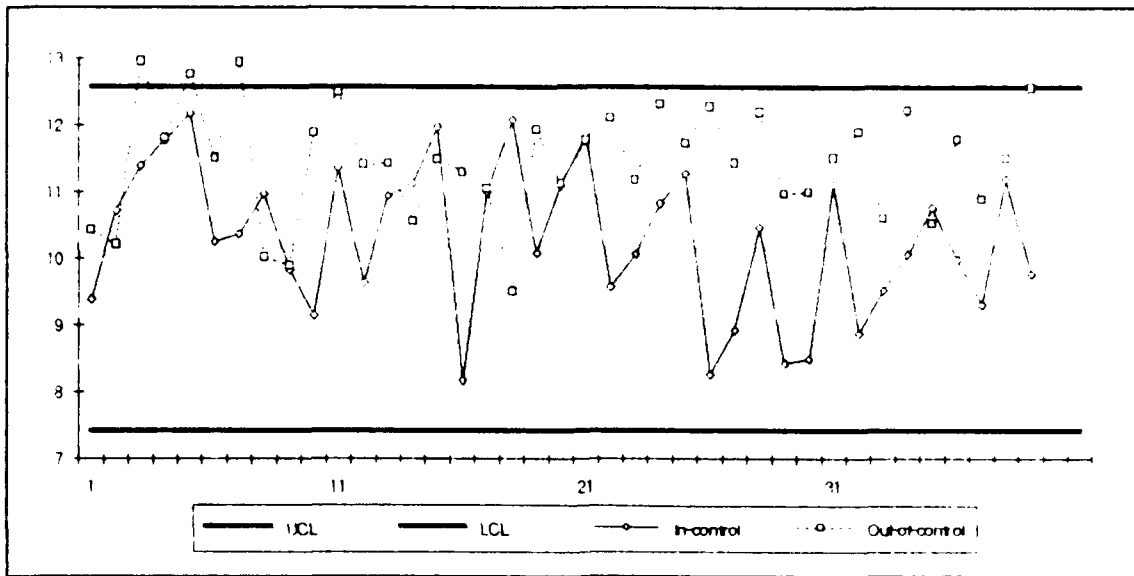
$$Pr (\tau_L < X' < \tau_U) = 0.10.$$

quality	$\mu$	$\sigma$	$\tau_L$	$\tau_U$	p
AQL	10	1	7.4242	12.5758	0.01
LTPD	11.2940	1	7.4242	12.5758	0.10

Table I

AQL in Table I stands for Acceptable Quality Level and is the desired maximum probability of stopping the production process if it is operating at the acceptable level. LTPD stands for Lot Tolerance Percent Defective and is the minimum probability of rejecting the process when it is operating at the unacceptable level. Now using  $n=39$  and  $c=1$ , we generate 39 samples from the process for  $X$  and  $X'$  (Figure 17).

From Figure 17 we can see that 4 values exceed the tolerance level for the out-of-control process ( $X'$ ) and that all of the values fall within the tolerance limits for the in-control-process. The graph shows both processes together and can help the decision maker visualize the differences between them.



**Figure 17:** Process Control Chart

Figure 18 shows these results, for a different replication, in tabular format. In this figure C-Lot stand for the lot produced by the in-control-process and OOC Lot stands for the lot produced by the out-of-control process. The Defects column shows how many items exceeded the tolerance limits for each respective lot and the Lot column provides a recommendation based on the established decision rule, reject if  $c > 1$ .

	Defects	Lot
C-Lot	1	Accept
OOC Lot	3	Reject

**Figure 18:** Accept / Reject Table

We repeated this process 1,000 times and obtained the results shown in Figure 19.

	Total Accept	Total Reject
C-Lot	948	52
OOC_Lot	81	919

**Figure 19:** Result of 1000 Trials

Expected values for this process can be calculated from the equations:

$$\begin{aligned}
 \text{Producer's risk} &= 1 - \sum_{x=0}^{\dot{1}} \binom{39}{x} 0.01^x 0.99^{39-x} = 0.0581 \\
 \text{Consumer's risk} &= \sum_{x=0}^{\dot{1}} \binom{39}{x} 0.10^x 0.90^{39-x} = 0.0876
 \end{aligned} \quad (7)$$

Table II shows the effect of increased correlation as  $\phi$  is systematically increased.

$\phi$	Producer's Risk	Consumer's Risk
Theoretical Values	.0581	.0876
0.0	.052	.081
0.1	.059	.110
0.3	.064	.128
.05	.072	.170
0.7.	.087	.267
0.9	.082	.496

Table II.

For the  $\phi=0$  case, the computer results are close to the theoretical values, as they should be. However, as you change the  $\phi$  values to  $\phi=0.1, 0.3, 0.5, 0.7, 0.9$  you increase the correlation in the process while keeping the process variance as is (i.e.  $\sigma^2=1$ ), both the *producer's risk* and the *consumer's risk* increasingly differ from the i.i.d case. As seen in Table II the correlation in the process causes a problem, the risk estimators are significantly different from the theoretical expectation of the normal i.i.d. case.

### C. CONCLUSION

This chapter has demonstrated how correlation can have a negative impact on sampling plans. The decision maker needs a tool to estimate the risk associated with such plans. Spreadsheets are an excellent tool that can provide the calculation power necessary and the graphical ability to help the decision maker better understand the problem. Complex relationships can be clarified and the manager is provided with graphs and tables to help him to draw conclusions that may not be otherwise obvious.



## VI CONCLUDING REMARKS

This thesis shows through the use of several diverse logistics problems that spreadsheets can provide a good foundation for logistics decision support systems. The availability of the personal computer and spreadsheet programs make them a logical choice for such systems. Further, the simplicity of their operation and the powerful output options available add greatly to their value for this purpose.

We have shown that spreadsheet models can be developed to provide decision support for a broad range of logistics problems. Models were presented in the areas of inventory simulation, time series analysis, data correlation and process control, and the value of graphical presentation as an aid to decision making for each case was discussed. Finally we have shown that spreadsheet programs can provide the logistics decision maker with the tools he needs to develop programs himself to solve the daily problems which confront him. The value of spreadsheet analysis as a foundation for logistics decision support lies more in its ready availability and simplicity than in its computational power. Custom-made programs, professionally developed to handle specific problems would likely be far more capable and could provide

far more elaborate graphical presentations. While this might be preferable from a scientific point of view it would be difficult to implement.

The cost and difficulty involved in the development of specialty programs would severely limit the applications which could be automated. The logistician needs systems that allow him to quickly model a system in which he is interested and obtain timely results to help him make decisions. While spreadsheets may not provide the most elegant decision support system they certainly provide sufficient power and flexibility support many of the decisions that confront a logistician on a daily basis.

The graphical options available in spreadsheet packages offer the decision maker with a broad array of display options. These presentations, previously costly, time consuming and difficult to obtain are now available to the user with a few keystrokes. Spreadsheet programming offers the logistician a desktop solution to the complex logistics problems that confront him on a daily basis.

## APPENDIX A INVENTORY MODEL CODE

### A. MAIN MACRO

#### **function NewValues**

```
WelcomeAns =DIALOG.BOX(dWelcomeScreen)
            = IF(WelcomeAns=8,GOTO(B5),)
            = IF(WelcomeAns=FALSE,CLOSE(FALSE))
InputAns =DIALOG.BOX(dInputSheet)
            = IF(InputAns=FALSE,GOTO(B1))
            =OPEN("main.xls")
            =OPEN("chart1.xlc")
            =RUN(Initialization)
            =FORMULA(DAY(NOW())&" / "&MONTH(NOW())&" /
                    "&YEAR(NOW())&"("&HOUR(NOW())&":"&MINUTE
(NOW())&")",R40)
            =RUN(Headers)
            =RUN(Repeater)
            =RUN(PlotNetInv)
            =RUN(PlotNetInvUpdate)
FinalAns =DIALOG.BOX(dFinalAns)
            =RUN(FinalScreenAns)
            =RETURN()
```

#### **function FinalScreenAns**

```
=IF(FinalAns=1)
= ACTIVATE("chart1.xlc")
= UNHIDE("chart1.xlc")
Keep changing= ON.KEY("~","work.xlm!R47c2")
            = PAUSE0
            = ON.KEY("~",)
            = HIDE0
            = GOTO(B15)
            =ELSE.IF(FinalAns=10)
```

```

=ELSE.IF(FinalAns=11)
= ACTIVATE("main.xls")
= UNHIDE("main.xls")
= ON.KEY("~","work.xlm!R47c2")
= PAUSE()
= ON.KEY("~",)
= HIDE()
= GOTO(B15)
=ELSE.IF(FinalAns=12)
=ELSE.IF(FinalAns=13)
=ELSE.IF(FinalAns=FALSE)
=RUN(CloseAllWorkFiles)
=END.IF()
=RETURN()

```

**function MacContinue**

```

=RESUME(1)
=RETURN()

```

**function CloseAllWorkFiles**

```

=ACTIVATE("chart1.xls")
=SAVE.AS("D:\THESIS\invmod\LASTWORK.XLC",
1,"",FALSE,"",FALSE)
=ACTIVATE("MAIN.XLS")
=SAVE.AS("D:\THESIS\invmod\LASTWORK.Xls",
1,"",FALSE,"",FALSE)
=ACTIVATE("LASTWORK.XLS")
=SELECT("R1c1:R"&WORK.XLM!Repetition&"c21", "R1C1")
=SELECT("R1:R16384")
=CLEAR(3)
=SELECT("R1C1")
=SAVE.AS("D:\THESIS\invmod\MAIN.XLS",1,"", FALSE,"",
FALSE,
=CLOSE()
=ACTIVATE("LASTWORK.Xlc")
=SAVE.AS("D:\THESIS\invmod\chart1.xls",
1,"",FALSE,"",FALSE)
=CLOSE(TRUE)
=RETURN()

```

**function Initialization**

```
=ACTIVATE("MAIN.XLS")
=SELECT("main.xls!R1c1:R"&WORK.XLM!
    Repetition&"c21","R1C1")
=CLEAR(3)
=ACTIVATE("WORK.XLM")
=FORMULA(0,X3)
=FORMULA(1,Repetition)
=FORMULA(0,AC1)
=FORMULA("",X5:AA7)
=FORMULA("",X9)
=SELECT(WORK.XLM!AC3:AE195)
= CLEAR(3)
=FORMULA(99999999999,X7)
=FORMULA(99999999999,X6)
=SELECT(WORK.XLM!AC3:AE398)
= CLEAR(3)
=FORMULA(0,AA5)
=FORMULA(0,AA6)
=FORMULA(0,AA7)
=RETURN()
```

**function Headers**

```
=FORMULA("Event",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!K2)
=FORMULA("Event",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!L1)
=FORMULA("Time",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!L2)
=FORMULA("Sys",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!M1)
=FORMULA("Recpt",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!M2)
=FORMULA("Cust",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!N1)
=FORMULA("Req",'D:\THESIS\PART1\INVMOD\MAIN.
    XLS'!N2)
```

```

=FORMULA("Sys",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!O1)
=FORMULA("Order",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!O2)
=FORMULA("Qty",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!P1)
=FORMULA("Iss",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!P2)
=FORMULA("Qty",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!Q1)
=FORMULA("BO",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!Q2)
=FORMULA("Qty",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!R1)
=FORMULA("OH",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!R2)
=FORMULA("Qty",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!S1)
=FORMULA("OO",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!S2)
=FORMULA("Net",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!T1)
=FORMULA("Inv",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!T2)
=FORMULA("Inv",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!U1)
=FORMULA("Pos",'D:\THESIS\PART1\INVMOD\MAIN.
  XLS'!U2)
=FORMULA("Beg
Bal",'D:\THESIS\PART1\INVMOD\MAIN. XLS'!K4)

=FORMULA(VALUE(WORK.XLM!OnHand),'D:\THESIS\PART1\
  INVMOD\MAIN.XLS'!R4)
=FORMULA(VALUE(WORK.XLM!OnOrder),'D:\THESIS\
  PART1\INVMOD\MAIN.XLS'!S4)

=FORMULA(VALUE(WORK.XLM!OnBackorder),'D:\THESIS\
  PART1\INVMOD\MAIN.XLS'!Q4)
=FORMULA('D:\THESIS\PART1\INVMOD\MAIN.XLS'!R4
  'D:\THESIS\PART1\INVMOD\MAIN.XLS'!Q4,'D:\
  THESIS\PART1\INVMOD\MAIN.XLS'!T4)
=FORMULA('D:\THESIS\PART1\INVMOD\MAIN.XLS'!R4
  'D:\THESIS\PART1\INVMOD\MAIN.XLS'!Q4+'D:

```

```

\THESIS\PART1\INVMOD\MAIN.XLS"!S4,'D:\THESIS\
PART1\INVMOD\MAIN.XLS"!U4)
=SELECT(WORK.XLM!V1)
=RETURN()

```

**function CalcCustRec**

```

=IF(DemandDist=1)
= FORMULA(0,V1)
=ELSE.IF(DemandDist=2)
= FORMULA(VLOOKUP(RAND(),TEXTREF("r3c36:r"&
(AG2+4)&"c37"),2),V1)
=ELSE.IF(DemandDist=3)
= FORMULA(VLOOKUP(RAND(),TEXTREF("r3c36:r"&
(AG2+4)&"c37"),2),V1)
=ELSE.IF(DemandDist=4)
minmDD = DemandMean
maxmDD = DemandStdDev
= FORMULA("",B146)
= FOR("count",1,ABS(minmDD-maxmDD),1)
= FORMULA(B146&"rand()",B146)
= NEXT()
=IF(randseriesDD="",FORMULA("rand()",B146))
= FORMULA(RIGHT(B146,(LEN(B146)-1)),B146)
= FORMULA("="&TEXT(minmDD,0)&" ("&randseriesDD
&")",B144)
UDAnsDD =0+(RAND())
= FORMULA(UDAnsDD,V1)
randseriesDDrand()
=ELSE.IF(DemandDist>4)
= FORMULA(RAND(),V1)
=END.IF()
=IF(RecQtyDist=1)
= FORMULA(0,V2)
=ELSE.IF(RecQtyDist=2)
= FORMULA(VLOOKUP(RAND(),TEXTREF("r3c33:r"&
(AG2+4)&"c34"),2),V2)
=ELSE.IF(RecQtyDist=3)
= FORMULA(VLOOKUP(RAND(),TEXTREF("r3c33:r"&
(AG2+4)&"c34"),2),V2)
=ELSE.IF(RecQtyDist=4)

```

```

minmQD      =RecQtyMean
maxmQD      =RecQtyStdDev
            =FORMULA("",B169)
            =FOR("count",1,ABS(minmQD-maxmQD),1)
            =FORMULA(B169&"rand()",B169)
            =NEXT()
            =IF(randseriesDD="",FORMULA("rand()",B169))
            =FORMULA(RIGHT(B169,(LEN(B169)-1)),B169)
            =FORMULA("="&TEXT(minmQD,0)&"+"&randseriesQD
            &"),B167)
UDAnsQD      =2+(RAND()+RAND()+RAND()+RAND())
            =FORMULA(ROUND(UDAnsQD,0),V2)
randseriesQDrand()+rand()+rand()+rand()
            =ELSE.IF(RecQtyDist>4)
            = FORMULA(RAND(),V2)
            =END.IF()
            =FORMULA(V1+X3,X5)
            =FORMULA(V2,Y5)
            =FORMULA(" ",Z5)
            =FORMULA(1,AA5)
            =RETURN()

```

#### **function CalcSysReorder**

```

=IF(GET.CELL(5,TEXTREF(ADDRESS(Repetition
+3,21,,FALSE,"main.xls")))<=TypeInvR)
= FORMULA(X3,X6)
= FORMULA("",Y6)
= FORMULA(TypeInvQ+(TypeInvR-GET.CELL(5,
TEXTREF(ADDRESS(Repetition+3,21,,FALSE,
"main.xls")))),Z6)
= RUN(CalcPLT)
= FORMULA((PLTAnswer+X6),TEXTREF(ADDRESS
(AC1+3,29,1,TRUE,),TRUE))
= FORMULA(Y6,TEXTREF(ADDRESS(AC1+3,
30,1,TRUE,),TRUE))
= FORMULA(Z6,TEXTREF(ADDRESS(AC1+3,
31,1,TRUE,),TRUE))
= FORMULA(AC1+1,AC1)
=FORMULA(1,AA6)
=ELSE()
= FORMULA(9999999999,X6)

```



```

= FORMULA("",Y6)
= FORMULA("",Z6)
= FORMULA(0,AA6)
=END.IF()
=RETURN()

```

**subroutine     CalcPLT**

```

=IF(PLTDist=1)
=ELSE.IF(PLTDist=2)
=ELSE.IF(PLTDist=3)
=ELSE.IF(PLTDist=4)
=ELSE.IF(PLTDist=5)
= FORMULA(PLTMean,B216)
=ELSE.IF(PLTDist>5)
=END.IF()
PLTAnswer 5
=RETURN(PLTAnswer)

```

**function       CalcSysRecpt**

```

=GET.CELL(5,AC1)
=IF(AC1=0)
= FORMULA(99999999999,X7)
= FORMULA("",Y7)
= FORMULA("",Z7)
= FORMULA(0,AA7)
=ELSE()
=FORMULA.GOTO("work.xlm!R3C31")
=SELECT("R3C29:R20C31")
=SET.DATABASE()
=SORT(1,"R3C29",1)
= IF(X7>=9999999999,COPY(AC3:AE3,X7),GOTO(B241))
= FORMULA(1,AA7)
= FORMULA("",AC3)
= FORMULA("",AD3)
= FORMULA("",AE3)
= SELECT("R3C29:R200C31")
= SET.DATABASE()
= SORT(1,"r3c29",1)
= FORMULA(AC1-1,AC1)

```

=END.IF()  
=RETURN()

**function      Repeater**

=IF(AA5=0,RUN(CalcCustRec),)  
=IF(AA6=0,RUN(CalcSysReorder),)  
=IF(AA7=0,RUN(CalcSysRecpt),)  
=FORMULA(MIN(X5:X7),X9)  
=IF(X9=X5)  
= FORMULA(W5,TEXTREF(ADDRESS(X4+4,11,  
    ,, "main.xls"),TRUE))  
= FORMULA(X5,TEXTREF(ADDRESS(X4  
    +4,12,,, "main.xls"),TRUE))  
= FORMULA(Y5,TEXTREF(ADDRESS(X4+  
    4,14,,, "main.xls"),TRUE))  
=RUN(CalcCustRecSub)  
= FORMULA(X3+V1,X3)  
= FORMULA(Repetition+1,Repetition)  
= FORMULA(" ",X5)  
= FORMULA(" ",Y5)  
= FORMULA(" ",Z5)  
= FORMULA(0,AA5)  
=ELSE.IF(X9=X6)  
= FORMULA(W6,TEXTREF(ADDRESS(X4  
    +4,11,,, "main.xls"),TRUE))  
= FORMULA(VALUE(X6),TEXTREF(ADDRESS(X4  
    +4,12,,, "main.xls"),TRUE))  
= FORMULA(VALUE(Z6),TEXTREF(ADDRESS(X4  
    +4,15,,, "main.xls"),TRUE))  
= RUN(CalcSysReordSub)  
= FORMULA(X6,X3)  
= FORMULA(Repetition+1,Repetition)  
= FORMULA(9999999999,X6)  
= FORMULA(" ",Y6)  
= FORMULA(" ",Z6)  
= FORMULA(0,AA6)  
=ELSE.IF(X9=X7)  
= FORMULA(W7,TEXTREF(ADDRESS(X4+4,  
    11,,, "main.xls"),TRUE))  
= FORMULA(X7,TEXTREF(ADDRESS(X4+4,12  
    ,,, "main.xls"),TRUE))

```

= FORMULA(Z7,TEXTREF(ADDRESS(X4+4,13
    ,,"main.xls"),TRUE))
= RUN(CalcSysRecptSub)
= FORMULA(X7,X3)
= FORMULA(Repetition+1,Repetition)
= FORMULA(999999999999,X7)
= FORMULA(" ",Y7)
= FORMULA(" ",Z7)
= FORMULA(0,AA7)
=END.IF0
=IF(X3<EndTime,GOTO(B246),)
=RETURN()

```

**subroutine    CalcSysReordSub**

```

=FORMULA(GET.CELL(5,TEXTREF(ADDRESS
    (Repetition+3,17,,FALSE,"main.xls"))),
    TEXTREF(ADDRESS(Repetition+4,17,,FALSE.
        "main.xls"))))
=FORMULA(GET.CELL(5,TEXTREF(ADDRESS
    (Repetition+3,18,,FALSE,"main.xls"))),TEXTREF
(ADDRESS(Repetition+4,18,,FALSE,"main.xls")))
=FORMULA(GET.CELL(5,TEXTREF(ADDRESS
    (Repetition+3,19,,FALSE,"main.xls")))+GET.
    CELL(5,TEXTREF(ADDRESS(Repetition+4,15,
        ,FALSE,"main.xls"))),TEXTREF(ADDRESS(
        Repetition+4,19,,FALSE,"main.xls")))
=FORMULA(("="&ADDRESS(Repetition+4,18,
    ,FALSE,"main.xls")&" "&ADDRESS(Repetition
    +4,17,,FALSE,"main.xls")),TEXTREF(ADDRESS
    (Repetition+4,20,,FALSE,"main.xls")))
=FORMULA(("="&ADDRESS(Repetition+4,18,
    ,FALSE,"main.xls")&" "&ADDRESS(Repetition
    +4,17,,FALSE,"main.xls")&" "&ADDRESS(
    Repetition+4,19,,FALSE,"main.xls")),TEXTREF
    (ADDRESS(Repetition+4,21,,FALSE,"main.xls")))
=RETURN()

```

**subroutine    CalcCustRecSub**

```

=IF(GET.CELL(5,TEXTREF(ADDRESS(Repetition+3,
18,,FALSE,"main.xls")))>=GET.CELL
(5,TEXTREF(ADDRESS(Repetition+4,
14,,FALSE,"main.xls"))))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+4,14,,FALSE,"main.xls"))),
TEXTREF(ADDRESS(Repetition+4,16,
,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,17,,FALSE,"main.xls"))),
TEXTREF(ADDRESS(Repetition+4,17,
,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,18,,FALSE,"main.xls"))),
GET.CELL(5,TEXTREF(ADDRESS(Repetition+4,14,
,FALSE,"main.xls"))),TEXTREF(ADDRESS
(Repetition+4,18,,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,19,,FALSE,"main.xls"))),
TEXTREF(ADDRESS(Repetition+4,19,,FALSE,
"main.xls")))
=ELSE()
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,18,,FALSE,"main.xls"))),
TEXTREF(ADDRESS(Repetition+4,16,,FALSE,
"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,17,,FALSE,"main.xls")))
+GET.CELL(5,TEXTREF(ADDRESS(Repetition+4,14,
,FALSE,"main.xls")))-GET.CELL(5,
TEXTREF(ADDRESS(Repetition+3,18,
,FALSE,"main.xls"))),TEXTREF(ADDRESS
(Repetition+4,17,,FALSE,"main.xls")))
= FORMULA(0,TEXTREF(ADDRESS(Repetition+4,18,
,FALSE,"main.xls")))
=FORMULA(GET.CELL(5,TEXTREF(ADDRESS(Repetition+3,
19,,FALSE,"main.xls"))),TEXTREF(ADDRESS
(Repetition+4,19,,FALSE,"main.xls")))
=END.IF()
=FORMULA(("="&ADDRESS(Repetition+4,
18,,FALSE,"main.xls")&" "&ADDRESS
(Repetition+4,17,,FALSE,"main.xls")),

```

```

TEXTREF(ADDRESS(Repetition+4,20,,FALSE,
"main.xls")))
=FORMULA(("="&ADDRESS(Repetition+4,18,
,FALSE,"main.xls")&" "&ADDRESS
(Repetition+4,17,,FALSE,"main.xls")&
"+"&ADDRESS(Repetition+4,19,
,FALSE,"main.xls")),TEXTREF(ADDRESS
(Repetition+4,21,,FALSE,"main.xls")))
=RETURN()

```

**subroutine    CalcSysRecptSub**

```

=IF(GET.CELL(5,TEXTREF(ADDRESS(Repetition+4,
17,,FALSE,"main.xls")))>=GET.CELL(5,
TEXTREF(ADDRESS(Repetition+3,17,,FALSE,
"main.xls"))))
= FORMULA(0,TEXTREF(ADDRESS(Repetition+4,
17,,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS(
Repetition+3,18,,FALSE,"main.xls")))+GET.
CELL(5,TEXTREF(ADDRESS(Repetition+4,
13,,FALSE,"main.xls")))+GET.CELL(5,TEXTREF
(ADDRESS(Repetition+3,17,,FALSE,
"main.xls"))),TEXTREF(ADDRESS(Repetition+4,
18,,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,19,,FALSE,"main.xls")))+GET.CELL
(5,TEXTREF(ADDRESS(Repetition+4,13,
,FALSE,"main.xls"))),TEXTREF(ADDRESS
(Repetition+4,19,,FALSE,"main.xls")))
=ELSE()
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS
(Repetition+3,17,,FALSE,"main.xls")) -
GET.CELL(5,TEXTREF(ADDRESS(Repetition
+4,13,,FALSE,"main.xls"))),TEXTREF(
ADDRESS(Repetition+4,17,,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS(Repetition
+3,18,,FALSE,"main.xls"))),TEXTREF(ADDRESS
(Repetition+4,18,,FALSE,"main.xls")))
= FORMULA(GET.CELL(5,TEXTREF(ADDRESS(Repetition
+3,19,,FALSE,"main.xls")))+GET.CELL(5,
TEXTREF(ADDRESS(Repetition+4,13,,FALSE,

```

```

"main.xls"))),TEXTREF(ADDRESS(Repetition
+4,19,,FALSE,"main.xls")))
=END.IF()
=FORMULA(("="&ADDRESS(Repetition+4,18,,FALSE,
"main.xls")&" "&ADDRESS(Repetition+4,17,
,FALSE,"main.xls")),TEXTREF(ADDRESS
(Repetition+4,20,,FALSE,"main.xls")))
=FORMULA(("="&ADDRESS(Repetition+4,18,,FALSE,
"main.xls")&"-"&ADDRESS(Repetition+4,17,
,FALSE,"main.xls")&"+"&ADDRESS(Repetition
+4,19,,FALSE,"main.xls")),TEXTREF(ADDRESS
(Repetition+4,21,,FALSE,"main.xls")))
=RETURN()

```

#### **function PlotNetInv**

```

nmi
="main.xls!r4c12:r"&(WORK.XLM!Repetition
+4)&"c12,main.xls!r4c20:r"&(WORK.XLM!
Repetition+4)&"c20"
=ACTIVATE("MAIN.XLS")
=COPY()
=CREATE.OBJECT(5,"R1C1",21,7.5,"R15C9",27.75,
6.75,1,TRUE)
=CHART.WIZARD(TRUE,TEXTREF(WORK.XLM!nmi),3,8,2,
1,2,2,"Net Inventory","Time","Units","")
=CHART.WIZARD(TRUE,"MAIN.XLS!R4C12:R100C12,
R4C20:R100C20",3,8,2,1,2,2,"Net Inventory",
"Time","Units","")
=RETURN()

```

#### **function PlotNetInvUpdate**

```

=UNHIDE("MAIN.XLS Chart 1")
=WINDOW.MAXIMIZE()
PlotNetInvX="main.xls!r4c12:r"&(Repetition+4)&"c12"
PlotNetInvY="main.xls!r4c20:r"&(Repetition+4)&"c20"
=ACTIVATE("MAIN.XLS Chart 1")
=EDIT.SERIES(1,"NetInv",TEXTREF(WORK.XLM!
PlotNetInvX),TEXTREF(WORK.XLM!PlotNetInvY),,1)
=RETURN()

```

**function MakeDistributionTables**

```
=SELECT(WORK.XLM!AG:AK)
=CLEAR(3)
=SELECT(WORK.XLM!V1)
=IF(WORK.XLM!RecQtyDist=2,RUN(MakeDmdQty
    TabNormal),)
=IF(WORK.XLM!RecQtyDist=3,RUN(MakeDmdQty
    TabPoisson),)
=IF(WORK.XLM!DemandDist=2,RUN(MakeDmdFreq
    TabNormal),)
=IF(WORK.XLM!DemandDist=3,RUN(MakeDmdFreq
    TabPoisson),)
=RETURN()
```

**function MakeDmdQtyTabNormal**

```
=FORMULA(1,AG1)
=IF(T(TEXTREF("r"&AG1+1&"c33"))="",GOTO(B366),
    GOTO(B370))
= FORMULA(" ", "r"&AG1+1&"c33")
= FORMULA(" ", "r"&AG1+1&"c34")
= FORMULA(AG1+1,AG1)
= GOTO(B365)
=FORMULA("Normal Qty",AG1)
=FORMULA(0,AG2)
=FORMULA(0,AG3)
=FORMULA(0,AH3)
=FORMULA(AG2,"r"&AG2+4&"c34")
=FORMULA(NORMDIST(AG2,RecQtyMean,RecQty
    StdDev,TRUE),"r"&AG2+4&"c33")
=IF(VALUE(TEXTREF("r"&AG2+4&"c33"))=1,GOTO
    (WORK.XLM!B379),GOTO(WORK.XLM!B377))
=FORMULA(AG2+1,AG2)
=GOTO(B374)
=RETURN()
```

**function MakeDmdQtyTabPoisson**

```

=FORMULA(1,AG1)
=IF(T(TEXTREF("r"&AG1+1&"c33"))="",GOTO(B385)
,GOTO(B389))
= FORMULA(" ", "r"&AG1+1&"c33")
= FORMULA(" ", "r"&AG1+1&"c34")
= FORMULA(AG1+1,AG20)
= GOTO(B384)
=FORMULA("Poisson Qty",AG1)
=FORMULA(0,AG2)
=FORMULA(0,AG3)
=FORMULA(0,AH3)
=FORMULA(AG2,"r"&AG2+4&"c34")
=FORMULA(POISSON(AG2,RecQtyMean,TRUE),"r"
&AG2+4&"c33")
=IF(VALUE(TEXTREF("r"&AG2+4&"c33"))>=0.999999999,
GOTO(WORK.XLM!B398),GOTO(WORK.XLM!B396))
=FORMULA(AG2+1,AG2)
=GOTO(B393)
=RETURN()

```

#### **function MakeDmdFreqTabNormal**

```

=FORMULA(1,AJ1)
=IF(T(TEXTREF("r"&AJ1+1&"c36"))="",GOTO
(B404),GOTO(B408))
= FORMULA(" ", "r"&AJ1+1&"c36")
= FORMULA(" ", "r"&AJ1+1&"c37")
= FORMULA(AJ1+1,AJ1)
= GOTO(B403)
=FORMULA("Normal Freq",AJ1)
=FORMULA(0,AJ2)
=FORMULA(0,AJ3)
=FORMULA(0,AK3)
=FORMULA(AJ2,"r"&AJ2+4&"c37")
=FORMULA(NORMDIST(AJ2,DemandMean,Demand
StdDev,TRUE),"r"&AJ2+4&"c36")
=IF(VALUE(TEXTREF("r"&AJ2+4&"c36"))=1,GOTO
(WORK.XLM!B417),GOTO(WORK.XLM!B415))
=FORMULA(AJ2+1,AJ2)
=GOTO(B412)
=RETURN()

```



**function MakeDmdFreqTabPoisson**

```

=FORMULA(1,AJ1)
=IF(T(TEXTREF("r"&AJ1+1&"c36"))="",GOTO
(B423),GOTO(B427))
= FORMULA(" ","r"&AJ1+1&"c36")
= FORMULA(" ","r"&AJ1+1&"c37")
= FORMULA(AJ1+1,AJ1)
= GOTO(B422)
=FORMULA("Poisson Freq",AJ1)
=FORMULA(0,AJ2)
=FORMULA(0,AJ3)
=FORMULA(0,AK3)
=FORMULA(AJ2,"r"&AJ2+4&"c37")
=FORMULA(POISSON(AJ2,DemandMean,TRUE),"r"&
AJ2+4&"c36")
=IF(VALUE(TEXTREF("r"&AJ2+4&"c36"))=1,GOTO
(WORK.XLM!B436),GOTO(WORK.XLM!B434))
=FORMULA(AJ2+1,AJ2)
=GOTO(B431)
=RETURN()

```

**B. INPUT DIALOG BOX**

	3	9	822	353	Input Sheet
5	500	10			Run Date:
5	25	10			Item Name:
5	10	30			Initial Inventory Quantities
5	10	50			On Hand:
5	10	70			Requisition Frequency
5	10	90			Type Distribution:
5	10	110			Mean:
5	10	130			Standard Deviation:
5	10	150			Requisition Quantity
5	10	170			Type Distribution:
5	10	190			Mean:
5	10	210			Standard Deviation:
5	10	230			Stockout Information
5	10	250			Type Distribution:
5	10	270			Mean:
5	10	290			Standard Deviation:

5 10 310  
 5 405 70  
 5 405 90  
 5 405 110  
 5 215 50  
 5 419 50  
 5 405 130  
 5 405 150  
 5 405 170  
 5 405 190  
 5 405 210  
 5 405 230  
 5 405 250  
 5 405 270  
 5 405 330  
 5 611 270  
 1 475 327 88  
 2 575 327 88  
 24 675 327 88  
 6 135 7 332  
 209 585 7 190  
 6 122 47 60  
 6 324 47 60  
 6 567 47 60  
 21 182 87 160  
 8 182 107 60  
 8 182 127 60  
 21 182 167 160  
 8 182 187 60  
 8 182 207 60  
 21 182 247 160  
 8 182 267 60  
 8 182 287 60  
 8 182 307 100  
 21 600 87 160  
 8 600 107 160  
 8 600 127 60  
 8 600 147 60  
 21 600 187 160  
 8 600 207 60  
 8 600 227 60  
 8 510 267 100  
 8 710 267 100

Cost per Stockout:  
 Type of Inventory Replenishment  
 System:  
 Frequency:  
 On Order:  
 On Backorder:  
 Q  
 R  
 Procurement Lead Time  
 Type Distribution:  
 Mean:  
 Standard Deviation:  
 Time Period to Record Data  
 Start Time:  
 Options:  
 End Time:  
 OK  
 Cancel  
 Help

Component XYZ  
 4 / 5 / 1993  
 72  
 0  
 0  
 r1c20:r10c204  
 0.2  
 0.8  
 R1C20:R10C204  
 2  
 6  
 R1C20:R10C204  
 0.8  
 0.2  
 100  
 R1C21:R4C211  
 4  
 54  
 18  
 R1C20:R10C205  
 5  
 0  
 1  
 5

## APPENDIX B AR(1) MODEL

### A. MAIN MACRO

**command NewValues**      **Generates the basic table**

```
=HIDE()
=OPEN("working.xls")
-0.9 =FORMULA(A4,'D:\THESIS\WORKAR1\WORKING.XLS'!C2)
=HIDE()
=OPEN("graphps.xls")
=HIDE()
=UNHIDE("GRAPHPS.XLS Chart 1")
=FOR("how",1,100,1)
=ACTIVATE("WORKING.XLS")
=CALCULATE.NOW()
= FOR("counter2",2,10,1)
= IF(counter2=4,GOTO($B$22))
= IF(counter2=5,GOTO($B$22))
= IF(counter2=6,GOTO($B$22))
= IF(counter2=8,GOTO($B$22))
= IF(counter2=9,GOTO($B$22))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r27c"&
    TEXT(counter2,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-3,0)))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r31c"&
    TEXT(counter2,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-2,0)))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r35c"&
    TEXT(counter2,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-1,0)))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r39c"&
    TEXT(counter2,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-0,0)))
= NEXT()
=NEXT()
=RETURN()
```

## B. INPUT DIALOG CODE

			492	135	
5	259	3			Parameter Box
5	146	38			Mu Value
5	127	59			Alpha Value
5	12	79			Number of Values required:
5	30	100			Number of Runs Desired:
8	233	34	96		0
8	233	55	96		0
7	233	76	96		0
7	233	97	96		10
1	369	29	88	OK	
2	373	59		Cancel	

## C. INPUT WORKSHEET

Alpha 0.9000  
Mu 0.0000

Random Table	16 As Is	By 2's	By 32's	By 256's
0.728039015	0.7280	0.5107	1.0500	0.3122
0.293417432	0.2934	0.5315	-0.3261	0.1452
0.702409882	0.7024	1.1422	0.2889	0.3244
0.360505867	0.3605	1.2907	0.3439	0.0888
1.146961164	1.1470	1.4165	1.4928	-0.2324
1.137490137	1.1375	1.8650	-1.2448	-0.0975
1.109449402	1.1094	1.3780	0.2834	0.1256
1.471979222	1.4720	1.2310	0.6098	-0.4336
1.347174986	1.3472	1.1471	0.6800	-0.6756
1.485807775	1.4858	0.6101	1.0451	-0.0360
2.153198621	2.1532	0.6677	0.9943	-0.0656
1.576787242	1.5768	0.6526	0.9129	-0.0856
1.228616318	1.2286	1.2237	-0.7779	0.0710
1.527331639	1.5273	1.6476	0.1417	0.0877
1.5031517	1.5032	0.9226	-1.0406	-0.2253
0.958851815	0.9589	0.5633	-0.7939	0.0878
0.899218644				

1.395063371	Mean		Mean	Mean	Mean
0.451370896	1.1707		1.0500	0.2287	-0.0381
0.768843129					
0.840207352					
0.49524308	St Dev		St Dev	St Dev	St Dev
0.716012221	0.1205		0.1063	0.2094	0.0651
0.589230872					
1.233709939					
1.213684499	UCL		UCL	UCL	UCL
1.581244831	1.3819		1.2364	0.5957	0.0760
1.713865226					
1.105883467					
0.739345047	LCL		LCL	LCL	LCL
0.721061958	0.9595		0.8637	-0.1383	-0.1521
0.405462228					
-0.502433799					
-0.605793575					
-0.595680786					

#### D. OUTPUT GRAPHS

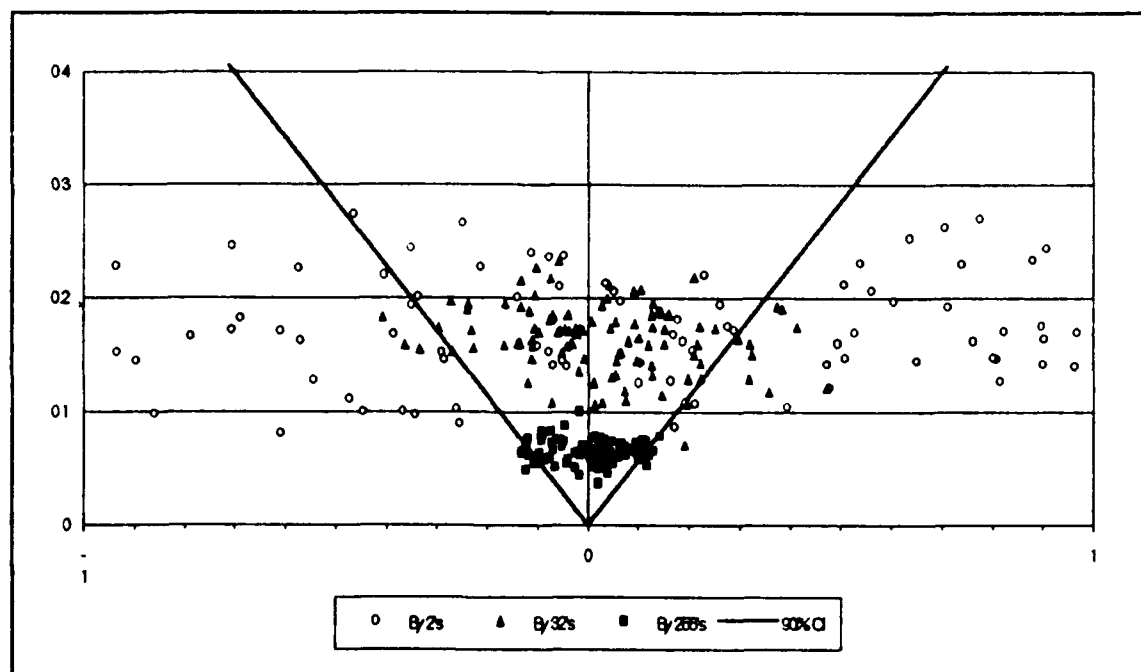
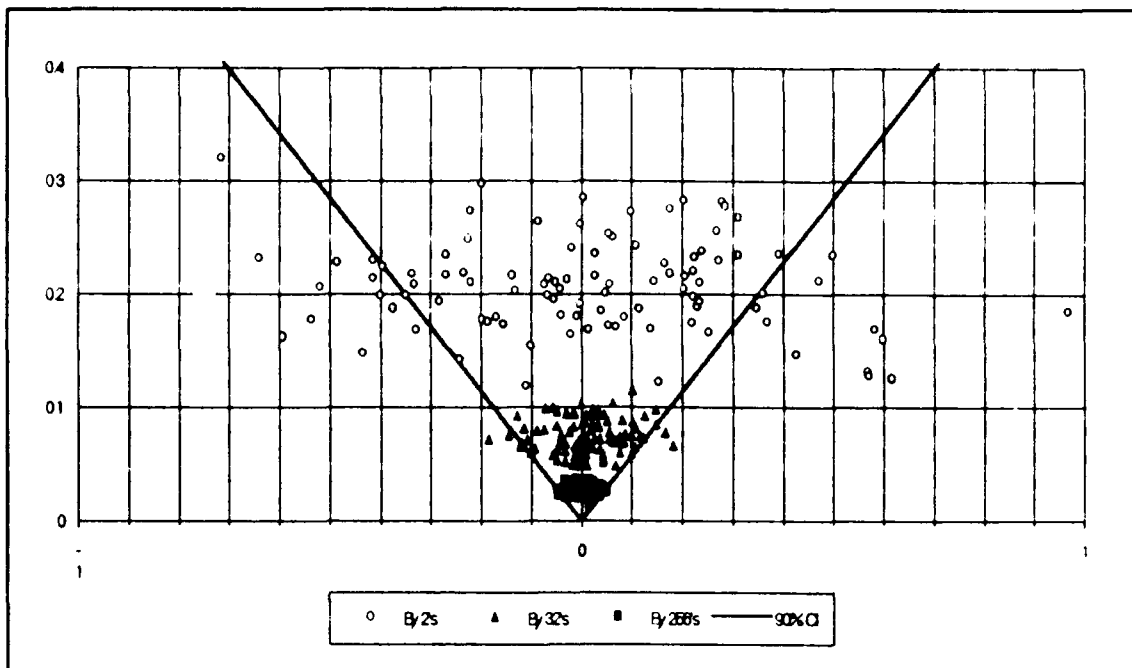
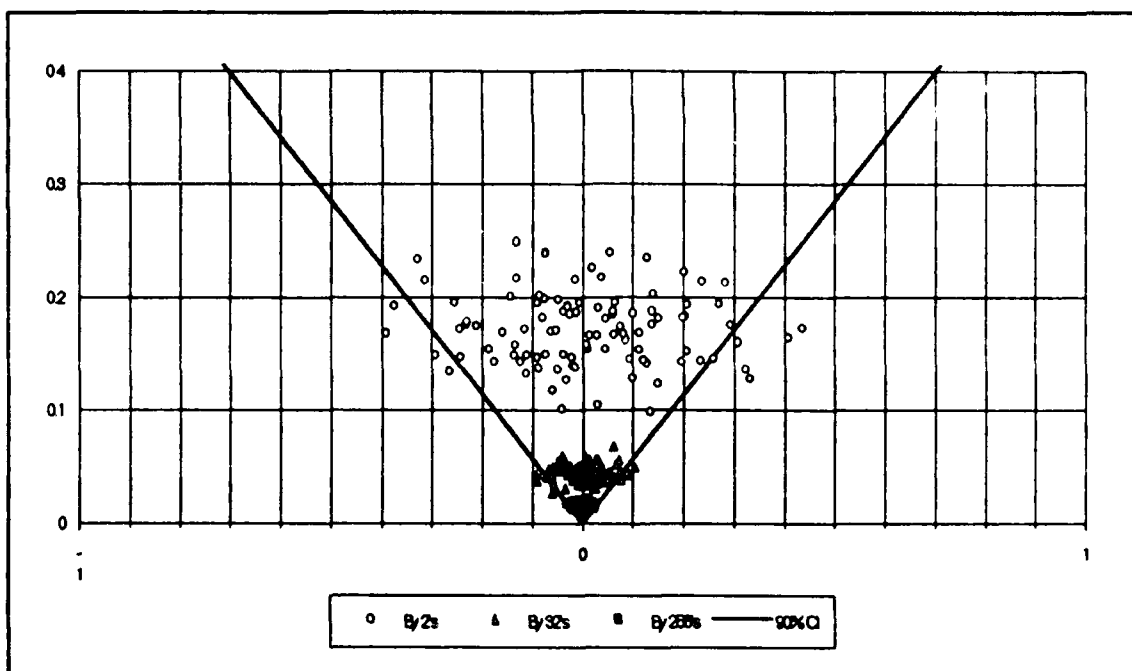


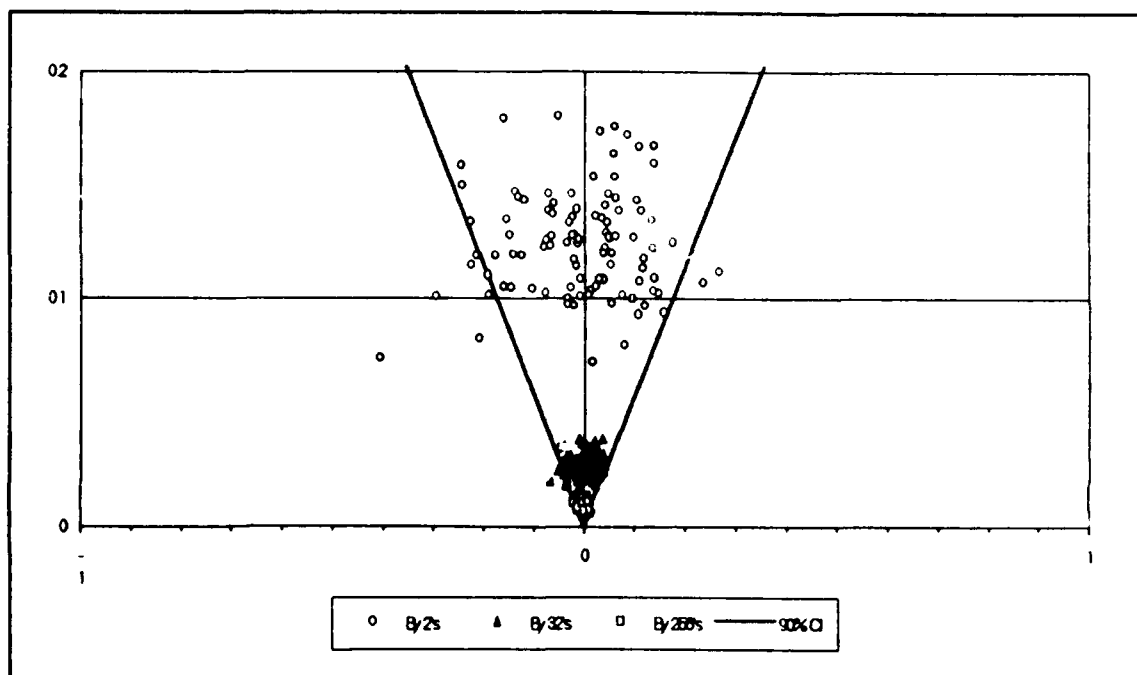
Figure 20: AR(1) Model,  $\phi=0.9$



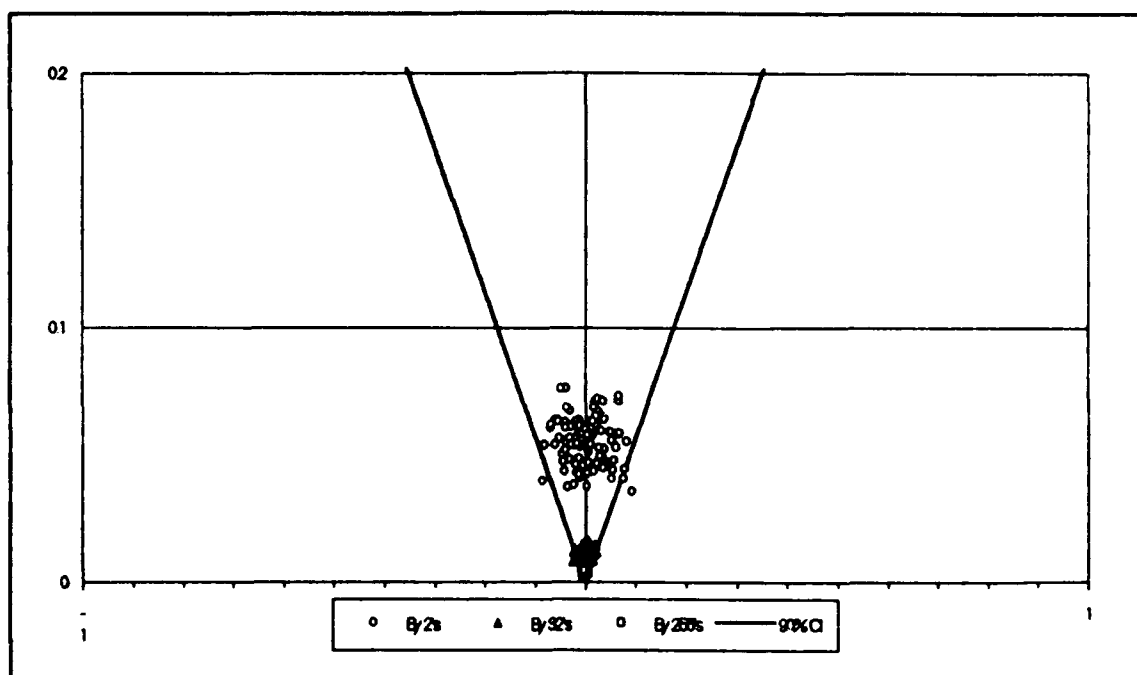
**Figure 21:** AR(1) Model,  $\phi=0.5$



**Figure 22:** AR(1) Model,  $\phi=0.0$



**Figure 23:** AR(1) Model,  $\phi = -0.5$



**Figure 24:** AR(1) Model,  $\phi = -0.9$

## APPENDIX C EAR(1) MODEL

### A. MAIN MACRO

**command NewValues** Generates the basic table

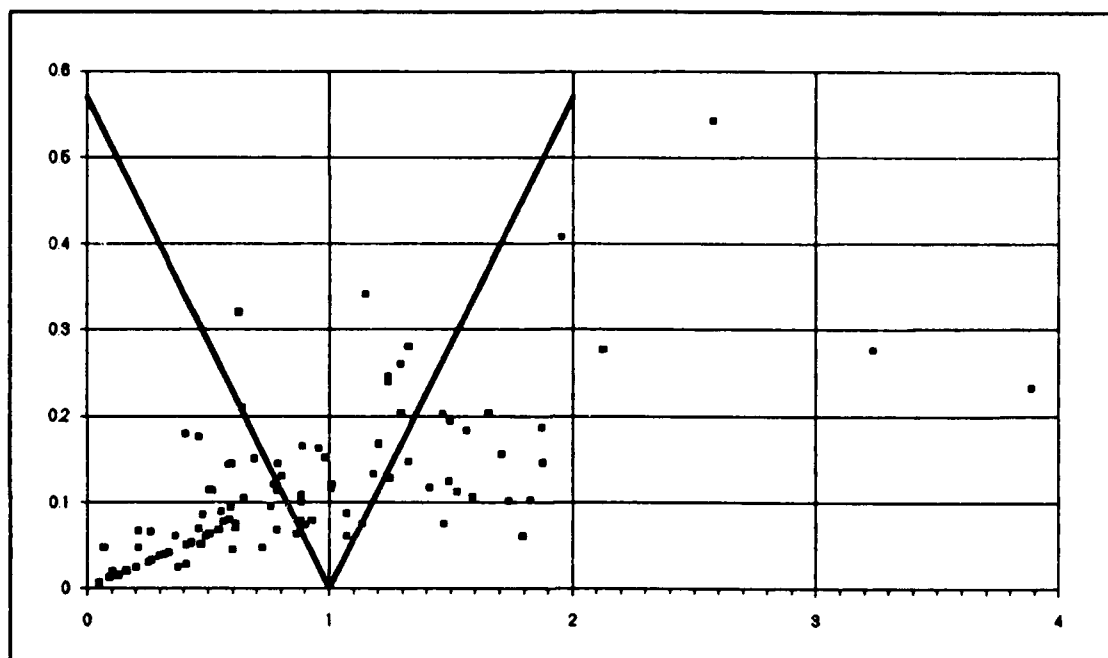
```
=HIDE()
=OPEN("working.xls")
-0.9 =FORMULA(A4,'D:\THESIS\WORKEAR1\WORKING.XLS'!D2)
=HIDE()
=OPEN("graphps.xls")
=HIDE()
=UNHIDE("GRAPHPS.XLS Chart 1")
=FOR("how",1,100,1)
=ACTIVATE("WORKING.XLS")
=CALCULATE.NOW()
= FOR("counter2",2,10,1)
= IF(counter2=4,GOTO($B$22))
= IF(counter2=5,GOTO($B$22))
= IF(counter2=6,GOTO($B$22))
= IF(counter2=8,GOTO($B$22))
= IF(counter2=9,GOTO($B$22))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r27c"&
    TEXT(counter2+1,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-3,0)))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r31c"&
    TEXT(counter2+1,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-2,0)))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r35c"&
    TEXT(counter2+1,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-1,0)))
=FORMULA(GET.CELL(5,TEXTREF("working.xls!r39c"&
    TEXT(counter2+1,0))),TEXTREF("graphps.xls!r"&
    TEXT(how+4,0)&"c"&TEXT(((counter2-1)*4)-0,0)))
= NEXT()
=NEXT()
=RETURN()
```



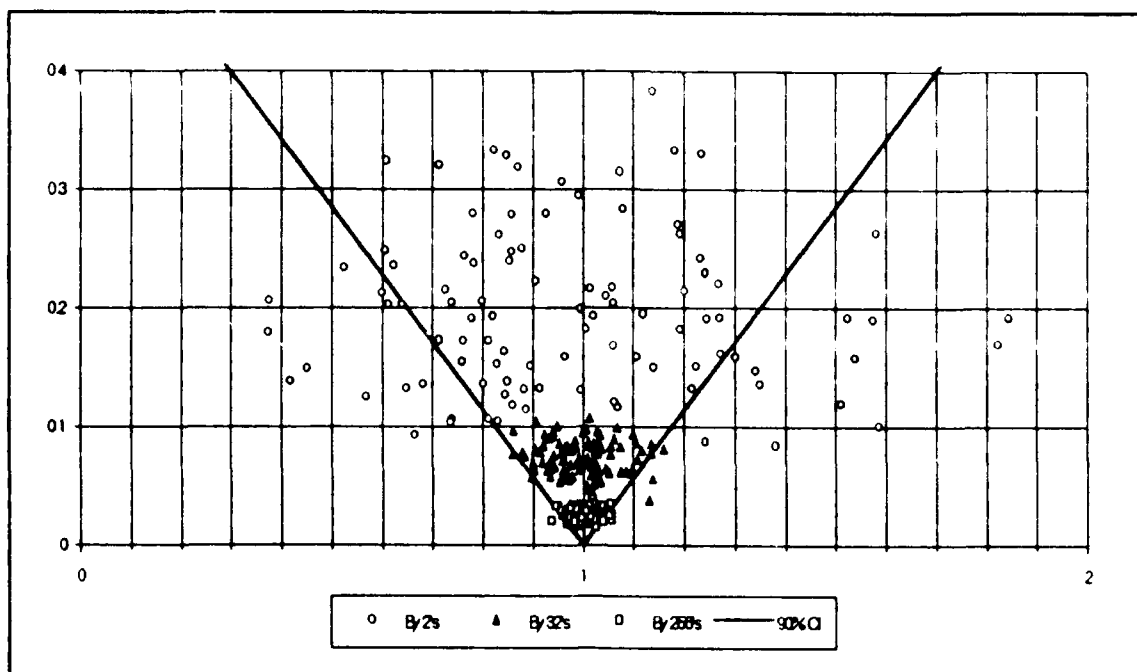
## B. INPUT DIALOG BOX

			492 135	
5	259	3		Parameter Box
5	146	38		Mu Value
5	127	59		Alpha Value
5	12	79		Number of Values required:
5	30	100		Number of Runs Desired:
8	233	34	96	0
8	233	55	96	0
7	233	76	96	0
7	233	97	96	10
1	369	29	88	OK
2	373	59		Cancel

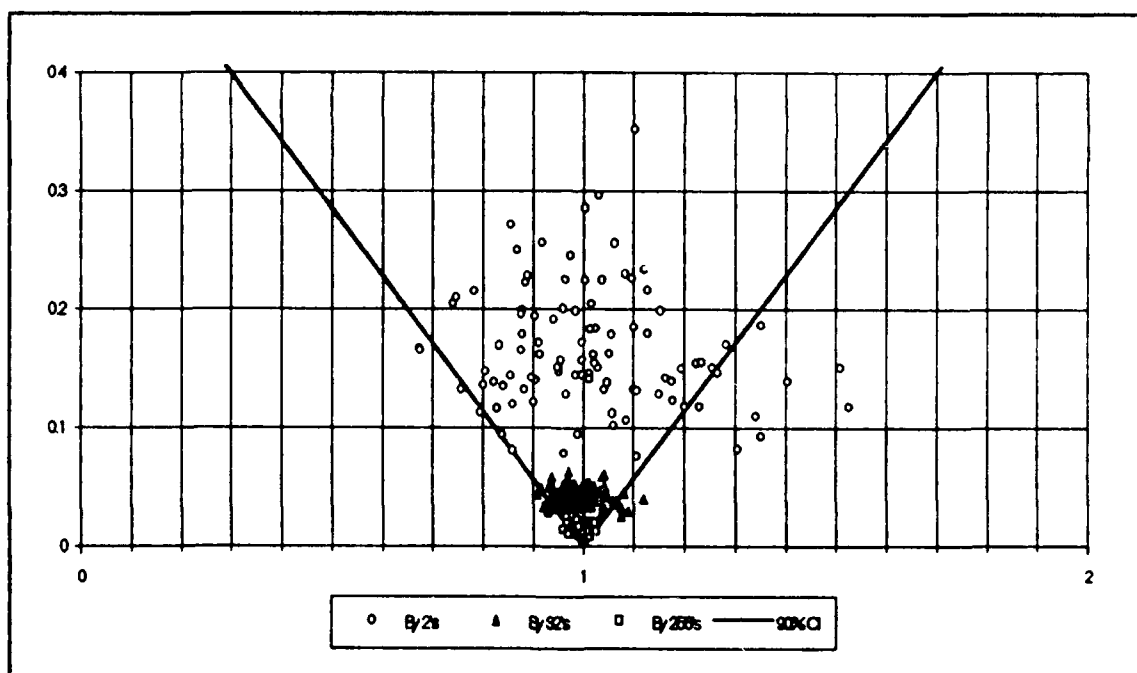
## C. OUTPUT GRAPHS



**Figure 25:** EAR(1) Model,  $\phi=0.9$



**Figure 26:** EAR(1) Model,  $\phi=0.5$



**Figure 27:** EAR(1) Model,  $\phi=0.0$

## APPENDIX D M/M/1 QUEUING MODEL

### A. MAIN MACRO

**command NewValues**      **Generates the basic table**

```
=DIALOG.BOX(ParameterBox)Calls entry Dialog Box
=HIDE()
=OPEN("graphps.xls")
=WINDOW.MAXIMIZE()
=HIDE()
=UNHIDE("graphps.xls Chart 1")
=WINDOW.MAXIMIZE()
=FORMULA("°90",C106)
=SET.VALUE(Arate,0.8)
=SET.VALUE(Srate,1)
=SET.VALUE(RHO,Arate/Srate)
=SET.VALUE(U,RAND())
=FORMULA(SaveCounter+1,SaveCounter)
=OPEN("D:\thesis\mm1pro\WORKING.XLS")
=HIDE()
=FORMULA(IF(U<1-RHO,0,-LN((1-U)/RHO)/(Srate-
    Arate)),D:\THESIS\PART2\MM1PRO\WORKING.XLS!A5)
=FOR("counter",1,NumberOfValues,1)
=SET.VALUE(Ratio,Srate/(Srate+Arate))
=SET.VALUE(U,RAND())
=IF(U<Ratio)
= SET.VALUE(DUM,LN(U/Ratio)/Arate)
=ELSE()
= SET.VALUE(DUM,-LN((1-U)/(1-Ratio))/Srate)
=END.IF()
=FORMULA(IF(GET.CELL(5,TEXTREF("working.xls!r"&
    TEXT(counter+4,0)&"c1"))+DUM>=0,GET.
    CELL(5,TEXTREF("working.xls!r"&TEXT(counter
    +4,0)&"c1"))+DUM,0),TEXTREF("working.xls!r"&
    TEXT(counter+1+4,0)&"c1"))
=NEXT()
=FOR("counter",1,16,1)
= FORMULA(TEXTREF("working.xls!r"&counter+4&
    +c1"),TEXTREF("working.xls!r"&counter+4&"c2"))
```

```

=NEXT()
=FOR("counter",1,64,4)
= FORMULA((SUM(TEXTREF("working.xls!r"&counter
+4&"c1"):TEXTREF("working.xls!r"&counter
+4+3&"c1"))/4),TEXTREF("working.xls!r"&
((counter+3)/4)+4&"c4"))
=NEXT()
=FOR("counter",1,4096,256)
= FORMULA((SUM(TEXTREF("working.xls!r"&counter+4&
"c1"):TEXTREF("working.xls!r"&counter+4+255&
"c1"))/256),TEXTREF("working.xls!r"&((counter
+255)/256)+4&"c10"))
=NEXT()
=ACTIVATE("WORKING.XLS")
=SAVE.AS("D:\thesis\mm1pro\"&C106&TEXT
(SaveCounter,0)&".XLS",1,"",FALSE,"",FALSE)
=HIDE()
= FOR("counter2",2,10,1)
= IF(counter2=3,GOTO($B$67))
= IF(counter2=5,GOTO($B$67))=ACTIVATE(C106&
SaveCounter&".xls")
= IF(counter2=6,GOTO($B$67))
= FORMULA(500+"r31c"&TEXT(counter2,0),TEXTREF
("graphps.xls!r"&TEXT(SaveCounter+4,0)&"c"&
TEXT(((counter2-1)*4)-1,0)))
= IF(counter2=7,GOTO($B$67))=COPY()
= IF(counter2=8,GOTO($B$67))=ACTIVATE("GRAPHPS.XLS")
= IF(counter2=9,GOTO($B$67))=SELECT("R9C30")
=FORMULA(500+GET.CELL(5,TEXTREF(C106&TEXT
(SaveCounter,0)&".xls!r23c"&TEXT(counter2,
0))),TEXTREF("graphps.xls!r"&TEXT(SaveCounter
+4,0)&"c"&TEXT(((counter2-1)*4)-3,0)))=PASTE()
=FORMULA(GET.CELL(5,TEXTREF(C106&TEXT(SaveCounter,
0)&".xls!r27c"&TEXT(counter2,0))),TEXTREF(
"graphps.xls!r"&TEXT(SaveCounter+4,0)&"c"&
TEXT(((counter2-1)*4)-2,0)))=RETURN()
=FORMULA(500+GET.CELL(5,TEXTREF(C106&TEXT
(SaveCounter,0)&".xls!r31c"&TEXT(counter2,0))),
TEXTREF("graphps.xls!r"&TEXT(SaveCounter+4,
0)&"c"&TEXT(((counter2-1)*4)-1,0)))
=FORMULA(500+GET.CELL(5,TEXTREF(C106&TEXT
(SaveCounter,0)&".xls!r35c"&TEXT(counter2,

```

```

0))),TEXTREF("graphps.xls!r"&TEXT(SaveCounter
+4,0)&"c"&TEXT(((counter2-1)*4)-0,0)))
= NEXT0
=CLOSE0
=ACTIVATE("variance.xlm")
=CHANGE.LINK(C106&TEXT(SaveCounter,0)&".xls",
"working.xls",1)
=IF(SaveCounter<NumberOfRuns,GOTO(B16),GOTO(B74))
=ACTIVATE("graphps.xls")
=SAVE.AS(C106&"graph")
=CLOSE0
=UNHIDE("variance.xlm")
=FORMULA(0,E1)
=IF(C106="o90")=GET.CELL(5,C106)
= FORMULA("'o50",C106)
=SET.VALUE(Alpha,0.5)
=ELSE.IF(C106="o50")
= FORMULA("'o00",C106)
=SET.VALUE(Alpha,0)
=ELSE.IF(C106="o00")
= FORMULA("'o05",C106)
=SET.VALUE(Alpha,-0.5)
=ELSE.IF(C106="o05")
= FORMULA("'o09",C106)
=SET.VALUE(Alpha,-0.9)
=ELSE.IF(C106="o09")
= GOTO(B96)
=END.IF0
=GOTO(B11)
=RETURN0

```

#### **function GatherGraphsCollects data to graph**

```

=OPEN("graphps.xls")
=FOR("counter",1,20,1)
=OPEN("run"&TEXT(counter,0)&".xls")
=HIDE0
= FOR("counter2",2,10,1)
=FORMULA(GET.CELL(5,TEXTREF("run"&TEXT(counter,
0)&".xls!r23c"&TEXT(counter2,0))),TEXTREF
("graphps.xls!r"&TEXT(counter+4,0)&"c"& TEXT(((counter2-
1)*4)-3,0)))

```

```

=FORMULA(GET.CELL(5,TEXTREF("run"&TEXT(counter,
0)&".xls!r27c"&TEXT(counter2,0))),TEXTREF
("graphps.xls!r"&TEXT(counter+4,0)&"c"&TEXT(((counter2-
1)*4)-2,0)))
=FORMULA(GET.CELL(5,TEXTREF("run"&TEXT
(counter,0)&".xls!r31c"&TEXT(counter2,0))),
TEXTREF("graphps.xls!r"&TEXT(counter+4,0)&"c"&
TEXT(((counter2-1)*4)-1,0)))
=FORMULA(GET.CELL(5,TEXTREF("run"&TEXT(counter,
0)&".xls!r35c"&TEXT(counter2,0))),TEXTREF(
"graphps.xls!r"&TEXT(counter+4,0)&"c"&TEXT (((counter2-
1)*4)-0,0)))
=NEXT()
=ACTIVATE("run"&TEXT(counter,0)&".xls")
=SAVE()
=CLOSE()
=NEXT()
=RETURN()

```

## B. WORKSHEET CODE

Alpha 0.9000

Mu 0.0000

Random Table	16 As Is	By 2's	By 32's	By 256's
-1.6213224	-1.6213	-1.4210	-0.2349	0.2909
-1.22059446	-1.2206	-1.3036	1.3514	0.0425
-1.22776797	-1.2278	-1.0833	1.5340	-0.1360
-1.37944501	-1.3794	-1.1246	0.9038	0.1470
-1.05692687	-1.0569	-1.7648	0.0035	0.1148
-1.10964019	-1.1096	-1.3869	0.0013	0.2470
-1.35727592	-1.3573	-1.2725	-0.7909	0.1856
-0.89182913	-0.8918	-0.4076	-0.4414	-0.3072
-1.60931646	-1.6093	-0.0011	-0.1820	0.0135
-1.92031504	-1.9203	0.3013	0.6717	-0.1757
-1.70954238	-1.7095	0.2239	-0.0210	0.6186
-1.06421186	-1.0642	0.7504	0.6593	-0.1585
-0.81630659	-0.8163	0.5641	0.0683	-0.1766
-1.72862307	-1.7286	1.5646	-0.0374	-0.0256
-0.60097121	-0.6010	1.0992	0.1304	0.1585

-0.21415811	-0.2142	1.5037	-0.9496	0.0140
0.05586469				
-0.05816206	Mean	Mean	Mean	Mean
0.49801669	-1.2205	-0.2349	0.1667	0.0533
0.10453922				
0.13928514				
0.30846461	St Dev	St Dev	St Dev	St Dev
0.74561632	0.4527	1.1266	0.6963	0.2280
0.75513539				
0.54237701				
0.58584934	UCL	UCL	UCL	UCL
1.69281666	-1.0343	0.2284	0.4530	0.1471
1.43644438				
1.02604077				
1.17245402	LCL	LCL	LCL	LCL
1.74208695	-1.4067	-0.6982	-0.1197	-0.0405

## APPENDIX E PROCESS CONTROL

### A. MAIN MACRO

```
=ACTIVATE("sqc.xls")
=FORMULA(0,'D:\THESIS\PART3\SQC.XLS'!L5)
=FORMULA(0,'D:\THESIS\PART3\SQC.XLS'!M5)
=FORMULA(0,'D:\THESIS\PART3\SQC.XLS'!L6)
=FORMULA(0,'D:\THESIS\PART3\SQC.XLS'!M6)
=FOR("e",1,GET.CELL(5,'D:\THESIS\PART3\SQC.XLS'!F5),1)
=ACTIVATE("SQC.XLS")
=SELECT("R8C2:R46C2","R46C2")
=COPY()
=PASTE.SPECIAL(3,1,FALSE,FALSE)
=SELECT("R8C5:R46C5","R46C5")
=COPY()
=PASTE.SPECIAL(3,1,FALSE,FALSE)
= FORMULA(a,'D:\THESIS\PART3\SQC.XLS'!H2)
= IF('D:\THESIS\PART3\SQC.XLS'!K5=" Accept")
=FORMULA('D:\THESIS\PART3\SQC.XLS'!L5+1,'D:
\THESIS\PART3\SQC.XLS'!L5)
=FORMULA('D:\THESIS\PART3\SQC.XLS'!M5,'D:\THESIS
\PART3\SQC.XLS'!M5)
= ELSE.IF('D:\THESIS\PART3\SQC.XLS'!K5=" Reject")
=FORMULA('D:\THESIS\PART3\SQC.XLS'!L5,'D:\THESIS
\PART3\SQC.XLS'!L5)
=FORMULA('D:\THESIS\PART3\SQC.XLS'!M5+1,'D:\THESIS
\PART3\SQC.XLS'!M5)
= END.IF()
= IF('D:\THESIS\PART3\SQC.XLS'!K6=" Accept")
=FORMULA('D:\THESIS\PART3\SQC.XLS'!L6+1,'D:\THESIS
\PART3\SQC.XLS'!L6)
=FORMULA('D:\THESIS\PART3\SQC.XLS'!M6,'D:\THESIS
\PART3\SQC.XLS'!M6)
= ELSE.IF('D:\THESIS\PART3\SQC.XLS'!K6=" Reject")
=FORMULA('D:\THESIS\PART3\SQC.XLS'!L6,'D:\THESIS
\PART3\SQC.XLS'!L6)
=FORMULA('D:\THESIS\PART3\SQC.XLS'!M6+1,'D:\THESIS
\PART3\SQC.XLS'!M6)
= END.IF()
```



```

=ACTIVATE("SQC.XLS)
=SELECT("R3C17:R41C17")
=COPY()
=SELECT("R8C2")
=PASTE()
=SELECT("R3C18:R41C18")
=COPY()
=SELECT("R8C5")
=PASTE()
=NEXT()
=RETURN()

```

## B. WORKSHEET LAYOUT

alpha =	0	n =	39
mu =	10	c =	1
muprime	11.294	Type I	0
=			
UCL =	12.575	Type II	0
	8		
LCL =	7.4242	replication	1000
		s	

C-Process			OOC-Process	
1	12.85874	1	10.556325	0
2	9.596326	0	11.010028	0
3	10.86229	0	10.988479	0
4	10.37656	0	9.9330193	0
5	9.091763	0	12.887687	1
6	9.968722	0	11.901957	0
7	8.653495	0	10.138580	0
8	9.437004	0	11.083508	0
9	8.921389	0	10.998890	0
10	10.76948	0	11.373073	0
11	9.572391	0	12.284658	0
12	10.48235	0	13.550179	1

13	9.790235	0	11.245558	0
14	9.771188	0	10.306608	0
15	9.169772	0	11.854465	0
16	9.302067	0	11.571596	0
17	9.733919	0	12.037539	0
18	10.90400	0	8.8534499	0
19	9.556637	0	12.233179	0
20	11.06074	0	13.661551	1
21	10.00515	0	10.975206	0
22	10.52555	0	9.6877362	0
23	9.501144	0	10.868471	0
24	11.67813	0	9.5916219	0
25	10.02455	0	10.519138	0
26	8.752327	0	10.095880	0
27	9.574731	0	9.5511311	0
28	12.26949	0	12.792277	1
29	11.38882	0	13.311218	1
30	10.12019	0	10.658583	0
31	11.68452	0	9.5926170	0
32	8.923597	0	10.850681	0
33	11.22553	0	11.107024	0
34	11.84095	0	10.750177	0
35	9.157223	0	12.228601	0
36	10.32481	0	12.589703	1
37	12.28659	0	11.903699	0
38	11.21761	0	10.285871	0
39	9.451519	0	10.342672	0
rejection		1	rejection	6

## LIST OF REFERENCES

Anderson, D.R., Sweeney, D.J., Williams, T.A., *An Introduction to Management Science*, 6th ed., pp. 518-520, West Publishing Company, 1991

Nelson, B.L., "Estimating Acceptance-Sampling Plans for Dependent Production Processes", Working Paper Series No. 1989-009, Department of Industrial and Systems Engineering The Ohio State University, October 1, 1990

Kang, K., and Schmeiser, B., "Graphical Methods for Evaluating and Comparing Confidence-Interval Procedures," *Operations Research*, v. 38, pp. 546-553, May-June 1990

Pritaker, A.A.B., *Introduction to Simulation and SLAM II*, pp. 194-195, A Halstead Press Book, John Wiley & Sons, Inc., 1984

Sargent, R. G., Kang, K., and Goldsman, D., "An Investigation of Finite-Sample Behavior of Confidence Interval Estimators," *Operations Research*, v. 40, pp. 898-913, September-October 1992

Seila, A.F., and Banks, J., "Spreadsheet risk analysis using simulation," *Simulation*, v. 55, pp. 163-169, September 1990

### INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2. Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3. Professor Keebom Kang, Code AS/KK Naval Postgraduate School Monterey, CA 93943-5703	1
4. Professor K. L. Terasawa, Code AS/KT Naval Postgraduate School Monterey, CA 93943-5703	1
5. Dale W. Goeller 1914 Keeter Run Chesapeake, VA 23320-7432	2